

Ces quelques macros (pour UltraMacros TimeOut) facilitent l'insertion des commandes d'impression de SuperFonts, quand vous travaillez sur un fichier de Traitement de texte.

D'un seul geste, ces macros permettent d'INSERER les codes SuperFonts, que vous soyez en mode insertion ou en mode recouvrement, selon le même principe que les commandes Ctrl-S ou Ctrl-G d'Appleworks pour le soulignement ou pour les caractères gras.

Pour employer ces commandes,编译ez ce fichier avec le Macro Compiler, puis faites sa-Tab (Pomme-Fermée-Tab, ou Option-Tab sur le GS) : vous êtes prêt à insérer les commandes. Les macros se souviennent de la commande précédente ("début" ou "fin"), et insèrent le code approprié. Pour "effacer" cette mémoire et caler tous les codes sur "début", faire à nouveau sa-Tab.

start

----- caler les variables sur "début" -----

```
<tab>:<all : i = 0 : n = 0 : r = 0 : o = 0 : f = 0
  msg ' Les macros SuperFonts sont activées ' : wait 1000 :sa-k>! Cale les
variables
```

```
k:<all:msg "Rappel :
i=ital./o=ombré/r=relief/n=inversé/f=francisé/j=just.dr./g=just.gau." :hilight
1,23,8,23>!
```

----- code de justification à droite -----

```
j:<all C = peek $10F1 : insert : print "<jd>" : if C = 1 then oa-A : ctrl-x
elseoff sa-k>!
```

----- code de justification à gauche -----

```
g:<all:oa-0>JG<rtn esc>!
```

----- codes avec début ou fin -----

```
i:<all C = peek $10F1 : insert : i = i + 1 : if i = 2 then i = 0 : print "<if>"
: else print "<id>" : elseoff : if C = 1 then oa-A : ctrl-x elseoff sa-k>!
```

```
n:<all C = peek $10F1 : insert : n = n + 1 : if n = 2 then n = 0 : print "<nf>"
: else print "<nd>" : elseoff : if C = 1 then oa-A : ctrl-x elseoff sa-k>!
```

```
r:<all C = peek $10F1 : insert : r = r + 1 : if r = 2 then r = 0 : print "<rf>"
: else print "<rd>" : elseoff : if C = 1 then oa-A : ctrl-x elseoff sa-k>!
```

```
o:<all C = peek $10F1 : insert : o = o + 1 : if o = 2 then o = 0 : print "<of>"
: else print "<od>" : elseoff : if C = 1 then oa-A : ctrl-x elseoff sa-k>!
```

```
f:<all C = peek $10F1 : insert : f = f + 1 : if f = 2 then f = 0 : print "<ff>"
: else print "<fd>" : elseoff : if C = 1 then oa-A : ctrl-x elseoff sa-k>!
```

----- codes numériques -----

```
p:<all $8 = "<" : $9 = ' Taper le numéro de la police, puis RTN ' : goto ba-x>!
```

```
d:<all $8 = "<p" : $9 = ' Taper le numéro de l'image, puis RTN ' : goto ba-x>!
t:<all $8 = "<t" : $9 = ' Taper la valeur de tabulation, puis RTN ' : goto
ba-x>!
x:<all $8 = "<x" : $9 = ' Taper le numéro du registre des polices, puis RTN ' :
goto ba-x>!
```

----- routine pour les macros ci-dessus -----

```
<ba-x>:<all C = peek $10F1 : insert : print $8 : msg $9 : input : msg "" : print
">" : if C = 1 then oa-A : ctrl-x >!
```

end

----- quelques explications -----

La variable C est affectée de façon temporaire

Les variables i, n, o, r et f servent à indiquer si la macro doit insérer un code "début" ou un code "fin"

Explication d'une macro :

```
r:<all C = peek $10F1 : ( noter si le curseur est en mode insertion ou
recouvrement )
insert :                  ( imposer le mode insertion )
r = r + 1 :              ( ajouter un au compteur pour "relief" )
if r = 2 then r = 0 : print "<rf>" : ( si 2, début déjà fait. Ecrire code
fin, recalculer le compteur )
else print "<sb>" :      ( sinon écrire code début )
elseoff : if C = 1 then oa-A : ctrl-x elseoff : ( remettre au besoin le mode
recouvrement ; le ctrl-x rend le curseur visible )
elseoff sa-k>!          ( rappel des codes en bas d'écran )
```

N131 Référence en avant - (2) recalculs nécessaires
N132
Q131 Référence en avant - (2) recalculs nécessaires
Q132
T131 Référence en avant - (2) recalculs nécessaires
T132
N132 Référence à un domaine sans aucune Valeur
Q132 Référence à un domaine sans aucune Valeur
T132 Référence à un domaine sans aucune Valeur
K134 Référence en avant - (2) recalculs nécessaires
K135
N134 Référence en avant - (2) recalculs nécessaires
N135
Q134 Référence en avant - (2) recalculs nécessaires
Q135
T134 Référence en avant - (2) recalculs nécessaires
T135
K135 Référence à un domaine sans aucune Valeur
N135 Référence à un domaine sans aucune Valeur
Q135 Référence à un domaine sans aucune Valeur
T135 Référence à un domaine sans aucune Valeur
T140 Référence à une cellule sans Valeur
N143 Référence en avant - (3) recalculs nécessaires
Q134
Q135
Q143 Référence en avant - (2) recalculs nécessaires
T134
Q143 Référence en avant - (3) recalculs nécessaires
T134
T135
N144 Référence en avant - (2) recalculs nécessaires
Q131
N144 Référence en avant - (3) recalculs nécessaires
Q131
Q132
Q144 Référence en avant - (3) recalculs nécessaires
T131
T132
N145 Référence en avant - (2) recalculs nécessaires
Q137
Q145 Référence en avant - (2) recalculs nécessaires
T137
N146 Référence en avant - (2) recalculs nécessaires
Q141
Q146 Référence en avant - (2) recalculs nécessaires
T141
N148 Référence en avant - (2) recalculs nécessaires
T134
N148 Référence en avant - (3) recalculs nécessaires
T134
T135
Q148 Référence en avant - (2) recalculs nécessaires

T134
Q148 Référence en avant - (3) recalculs nécessaires
T134
T135
N149 Référence en avant - (3) recalculs nécessaires
T131
T132
Q149 Référence en avant - (3) recalculs nécessaires
T131
T132
N150 Référence en avant - (2) recalculs nécessaires
T137
Q150 Référence en avant - (2) recalculs nécessaires
T137
N151 Référence en avant - (2) recalculs nécessaires
T141
Q151 Référence en avant - (2) recalculs nécessaires
T141
Q156 Référence à un domaine sans aucune Valeur
N159 Référence en avant - (2) recalculs nécessaires
Q154
N160 Référence à une cellule sans Valeur
N161 Référence en avant - (2) recalculs nécessaires
Q156
Q161 Référence à un domaine sans aucune Valeur
N163 Référence en avant - (3) recalculs nécessaires
Q134
Q135
Q163 Référence en avant - (2) recalculs nécessaires
T134
Q163 Référence en avant - (3) recalculs nécessaires
T134
T135
N164 Référence en avant - (2) recalculs nécessaires
Q131
N164 Référence en avant - (3) recalculs nécessaires
Q131
Q132
Q164 Référence en avant - (3) recalculs nécessaires
T131
T132
N165 Référence en avant - (2) recalculs nécessaires
Q137
Q165 Référence en avant - (2) recalculs nécessaires
T137
N166 Référence en avant - (2) recalculs nécessaires
Q141
Q166 Référence en avant - (2) recalculs nécessaires
T141
N168 Référence en avant - (2) recalculs nécessaires
T134
N168 Référence en avant - (3) recalculs nécessaires
T134
T135

A95 Valeur non référencée
A96 Valeur non référencée
A97 Valeur non référencée
A98 Valeur non référencée
A99 Valeur non référencée
A100 Valeur non référencée
A101 Valeur non référencée
A102 Valeur non référencée
J102 Valeur non référencée
A103 Valeur non référencée
A104 Valeur non référencée
A105 Valeur non référencée
A106 Valeur non référencée
A107 Valeur non référencée
A108 Valeur non référencée
A109 Valeur non référencée
A110 Valeur non référencée
A111 Valeur non référencée
A116 Valeur non référencée
A117 Valeur non référencée
A118 Valeur non référencée
A119 Valeur non référencée
A120 Valeur non référencée
A121 Valeur non référencée
A122 Valeur non référencée
A123 Valeur non référencée
A124 Valeur non référencée
A125 Valeur non référencée
A126 Valeur non référencée
A127 Valeur non référencée
A128 Valeur non référencée
A129 Valeur non référencée
K130 Valeur non référencée
K136 Valeur non référencée
Q158 Valeur non référencée
Q160 Valeur non référencée
K181 Valeur non référencée
K182 Valeur non référencée

Q169 Référence en avant - (2) recalculs nécessaires
T134
Q169 Référence en avant - (3) recalculs nécessaires
T134
T135
N169 Référence en avant - (3) recalculs nécessaires
T131
T132
Q169 Référence en avant - (3) recalculs nécessaires
T131
T132
N170 Référence en avant - (2) recalculs nécessaires
T137
Q170 Référence en avant - (2) recalculs nécessaires
T137
N171 Référence en avant - (2) recalculs nécessaires
T141
Q171 Référence en avant - (2) recalculs nécessaires
T141
A53 Valeur non référencée
A54 Valeur non référencée
A55 Valeur non référencée
A56 Valeur non référencée
A57 Valeur non référencée
A58 Valeur non référencée
A59 Valeur non référencée
A60 Valeur non référencée
A61 Valeur non référencée
A62 Valeur non référencée
A63 Valeur non référencée
A64 Valeur non référencée
J72 Valeur non référencée
J73 Valeur non référencée
A74 Valeur non référencée
A75 Valeur non référencée
A76 Valeur non référencée
A77 Valeur non référencée
A78 Valeur non référencée
A79 Valeur non référencée
A80 Valeur non référencée
A81 Valeur non référencée
A82 Valeur non référencée
A83 Valeur non référencée
A84 Valeur non référencée
A85 Valeur non référencée
A86 Valeur non référencée
A87 Valeur non référencée
A88 Valeur non référencée
A89 Valeur non référencée
A90 Valeur non référencée
A91 Valeur non référencée
A92 Valeur non référencée
A93 Valeur non référencée
A94 Valeur non référencée

Grâce à ce macroprogramme, vous pourrez créer sans mal les listes "Link" dans vos Tableaux d'Import. Voici comment procéder :

1. Compilez cette macro avec Macro Compiler
2. Placez les fichiers Export et Import sur le Bureau
3. Allez au fichier Import, placez "Links"
4. Faites Option-B (Pomme-Pleine-B) et suivez les instructions.
5. Une fois les liens construits, faites Pomme-Pleine-L (Option-L) pour activer "Lier Cellules"

Variables Q R T U V W X Y Z

Chaines \$1, \$2, \$3

Macros all: B, M, L, F, Z asp: C, G, L

start

--- cette macro définit le fichier d'Import, et l'emplacement du premier lien ---

```
b:<all oa-q : msg ' Choisir le fichier IMPORT, faire Return ' : input rtn :  
q = peek $c54 : ( noter No du fichier )  
msg ' Curseur sur première cellule liée, puis Return ' : input :  
rtn : goto sa-M>!
```

----- boucle principale pour faire les liens -----

```
M:<asp : msg "" : oa-q : msg ' Choisir fichier EXPORT, ou Esc. pour quitter si  
terminé ' :
```

```
x = key : if x = 27 then rtn :  
msg "-----> Faites x-L pour importer les données <":  
poke 1630,64 : ( ce poke est une Pomme-Pleine MouseText ) stop elseoff  
  
print chr$ x : ( transmettre la touche ) input rtn  
r = peek $c54 ( noter No du fichier )  
msg ' Curseur sur première cellule à lier, puis Return ' : input  
  
posn w,x : sa-q ( cellule de départ )  
  
$3 = " 1ère cellule = " + $1 + ' : Curseur sur dernière cellule, puis RTN ' :  
oa-X : print "b" : (choisir "bloc")  
msg $3 : input : posn y,z : esc ( dernière cellule )  
$2 = screen 1,22,5 : u = 0 :  
  
if w = y then u = 1 else ( même ligne )  
if x = z then u = 2 elseoff ( même colonne )  
if u = 0 then $3 = $1 + " et " + $2 + " ne sont pas alignées "  
: msg $3 : bell : stop elseoff ( même ligne obligatoire )  
  
if u = 2 then v = y - w + 1 : else v = z - x + 1 : ( calc. maximum )  
elseoff : v = v - 1 : t = v :  
  
if w = x then u = 2 : v = 0 : elseoff ( cellule unique....)  
oa-n cell esc : goto sa-c>! ( nom fichier )
```

```

--- construire le chainage ---
c:<asp : sa-g ( référence cellule )

$2 = " " + $1 + "(" + $0 + ")" ; ( construire le lien)

if u = 2 then right else down elseoff ( passer à la cellule suivante)

oa-q : print q : rtn ( retour au fichier import)
print "A" : rtn ( forcer label )
oa-u oa-y ( oa-u pour ne pas déborder cellule)
print $2 : rtn down ( écrire le lien, aller à la ligne)

ifnot v = 0 then v = v - 1 : oa-q : print r : rtn : goto sa-c elseoff :
goto sa-m!

--- lire référence cellule sur écran ---
g:<asp L = 2 ( L = longueur référence )

begin
L = L + 1 :
$i = screen L,22,i : ifnot $i = " " then :
rpt elseoff : (cherche le premier espace après la référence)
L = L - 1 : $i=screen L,22,i : if $i=="": L=L-1 elseoff
$i = screen 1,22,L>! ( lire réf. cellule à l'écran )

--- executer Lier Cellules dans fichier actuel ---
L:<all $0 = "Lier Cellules" : ( nom de l'application )
oa-esc : sa-ctrl-f ( recherche $0 ; résultat dans z )

if z = 0 then msg $0 + ' introuvable ' :
bell : stop ( alors on arrête )

elseoff : rtn>! run it

Des deux macros sont le nouveau "module de recherche ZMM v2.0".

La recherche passe successivement en revue tous les menus TO, jusqu'à ce que
l'application soit trouvée, ou la liste épuisée.

<ctrl-f>:<all : y = peek 7794 * 256 + peek 7793 ( identif.le premier menu )
begin
sa-ctrl-z ( recherche avec ZMM v1.0 )
if z = 0 then tab ( négatif, changer de menu )
x = peek 7794 * 256 + peek 7793 ( identif. du menu actuel )
ifnot x = y then ( si nouveau menu, continue recherche )
rpt>!
NOTE: si x = y on est revenu au premier menu, il n'y a plus qu'à
abandonner...

<ctrl-z>:<all : z = 0 : find : z = 1>!
```

Module simple (pour liste unique), donne le résultat booléen dans Z : 0 = pas trouvé, 1 = trouvé.

ZMM Module de recherche TO v1.0 de Mark Munz
ZMM Module de recherche TO v2.0 de Randy Brandt

Intéresse tous les utilisateurs de TimeOut UltraMacros !

Voici un module de recherche conçu spécialement pour la version 2.0 de TimeOut à menus multiples.

Il appelle instantanément l'application TimeOut de votre choix, ou bien vous avertit qu'elle n'est pas disponible.

Astuce : Pour une recherche manuelle, supprimez la définition de \$0 dans la macro "\$" ci-après. Pour appeler une application TimeOut, faites Pomme-Ouverte-0 (zéro), tapez le nom de l'application, puis faites Pomme-Pleine-\$.

Si vous n'utilisez qu'une seule liste d'applications, <ctrl-z> suffit à la place de <sa-ctrl-f>.

start

\$:<all (<exemple de macro pour trouver et lancer une application TO>)

```
$0 = "Analyzer" :      ( nom de l'application )
oa-esc : sa-ctrl-f    ( recherche $0; résultat dans Z )
if z = 0 then msg $0 + ' n\'a pas été trouvé ' :
bell : stop           ( on arrête tout si la recherche est négative )
elseoff : rtn>!       ( démarrer l'application )
```

=====

Ces deux macros sont le nouveau "module de recherche ZMM v2.0".

La recherche passe successivement en revue tous les menus TO, jusqu'à ce que l'application soit trouvée, ou la liste épuisée.

```
<ctrl-f>:<all : y = peek 7794 * 256 + peek 7793 (<identif. le premier menu>
begin
  sa-ctrl-z                      ( recherche avec ZMM v1.0 )
  if z = 0 then tab                ( négatif, changer de menu )
  x = peek 7794 * 256 + peek 7793  ( identif. du menu actuel )
  ifnot x = y then                 ( si nouveau menu, continue recherche )
rpt>!
NOTE: si x = y on est revenu au premier menu, il n'y a plus qu'à abandonner...
```

```
<ctrl-z>:<all : z = 0 : find : z = 1>!
```

Module simple (pour liste unique), donne le résultat booléen dans Z : 0 = pas trouvé, 1 = trouvé.