

Un script AppleScript réversible pour se débarrasser du Dock

(A. Guiot)

AppleScript n'a pas accès directement aux propriétés du Dock. Il doit donner la main à l'interpréteur Unix (le shell) pour agir sur le Dock. La commande Unix en question se nomme "**defaults**". Petite leçon là-dessus.

La commande Unix "defaults" permet de lire et de modifier divers réglages de l'interface utilisateur, comme le Dock ou la barre des menus, ce que font certains softs utilitaires comme Onyx ou Xupport par exemple. On peut bien sûr écrire directement dans le Terminal, mais il est plus simple de composer un script AppleScript qui automatisera le travail. Dommage qu'Automator ne gère pas le Dock. AppleScript non plus d'ailleurs, sinon à travers la puissante commande "do shell script" qui donne la main à Unix.

La syntaxe de la commande "defaults" est la suivante :

"**defaults read <domaine> <propriété>**" permet la lecture d'une propriété d'un domaine.

"**defaults write <domaine> <propriété> <valeur>**" permet de modifier une propriété d'un domaine.

Pour le Dock, le domaine est "**com.apple.dock**". Vous pouvez lire les valeurs des propriétés dans le fichier "com.apple.dock.plist" situé dans le dossier Préférences de l'utilisateur (~ / Bibliothèque / Preferences / com.apple.dock.plist). Pour l'ouvrir, il faut disposer de l'application "Property List Editor" qui est fournie dans le pack des outils de développement d'Apple (Xcode : gratuit).

Comme les réglages du Dock se trouvent dans les Préférences système, il est facile de voir comment la modification d'un réglage à ce niveau se répercute dans le fichier com.apple.dock.plist.

Sept propriétés (sur une vingtaine) sont affectées par les Préférences système. Voici par exemple mes réglages actuels :

autohide	Boolean	Yes
largesize	Number	75.000000
launchanim	Boolean	Yes
magnification	Boolean	No
mineffect	String	genie
orientation	String	left
tilesize	Number	30.000000

Ça veut dire quoi ?

La première colonne est le nom de la propriété, la seconde le type de valeurs et la troisième la valeur que possède la propriété.

Trois types de valeurs ici :

Boolean = valeur booléenne, c-à-d "oui" ou "non" (dans le traitement informatique, la valeur est codée sur 1 bit : le "oui" vaut 1 et le "non" vaut 0)

Number = nombre (sans commentaire, sinon que le point fait office de virgule décimale)

String = chaîne (de caractères évidemment)

Voyons les propriétés au travers des réglages du Dock dans les Préférences système. Ouvrez-les pour bien comprendre.

1. Taille du Dock : propriété "**tilesize**".

C'est la taille d'une "tuile", donc d'un élément du Dock et par conséquent de sa largeur. Les valeurs imposées par le curseur vont de 16 à 128 (16.000000 à 128.000000 avec les décimales !).

2. Case à cocher **Agrandissement** : propriété "**magnification**".

Non cochée = "No" et cochée = "Yes".

3. **Agrandissement** (curseur) : propriété "**largesize**".

C'est le complément de taille que l'on observe quand le curseur passe sur les éléments du Dock. Les valeurs imposées par le curseur vont aussi de 16 à 128.

4. **Position** : propriété "**orientation**".

Trois valeurs : Gauche = "left", Droite = "right" et Bas = "bottom".

5. **Type de réduction** : propriété "**mineffect**".

Deux valeurs : effet génie = "genie" et effet d'échelle = "scale".

6. Case à cocher **Animation à l'ouverture des applications** : propriété "**launchanim**".

Non cochée = "No" et cochée = "Yes".

7. Case à cocher **Masquage/affichage automatique** : propriété "**autohide**".

Non cochée = "No" et cochée = "Yes".

Voilà. Si vous reprenez mes réglages du Dock de ci-dessus, vous pourrez lire que mon Dock est masqué et situé à gauche, que sa taille est 30 et donc assez petit, que l'agrandissement est désactivé, que l'icône d'une application s'anime lorsqu'on la lance, et qu'en cliquant sur le bouton jaune d'une fenêtre, celle-ci rejoint le Dock par effet génie.

Tout cela peut être aussi modifié par des scripts. Les avantages ? Automatiser et donc exécuter très rapidement des modifications du Dock, comme par exemple la miniaturisation et le déplacement d'un Dock "emmerdant" lorsque dans Photoshop, on travaille en plein écran là où il se trouve. Rien n'empêche non plus d'outrepasser les valeurs imposées par les Préférences système : une taille inférieure à 16 ou une position supérieure juste sous la barre des menus (valeur "top" non accessible dans les Préférences systèmes). Dans ce cas, pour afficher un Dock masqué, il faut placer exactement l'extrémité du pointeur-flèche sous la barre des menus.

Il existe aussi une valeur "suck" pour le type de réduction. Donc, en plus de l'effet génie et de l'effet d'échelle, il y a un effet aspirateur. Notez qu'il existe d'autres propriétés non accessibles par les Préférences système, comme mettre une ombre au Dock (propriété booléenne "showshadow") ou faire démarrer le Dock à partir d'un coin (propriété "pinning").

Essayez ceci : ouvrez le Terminal et tapez exactement ce qui suit (n'ayez pas peur). Si vous préférez, faites un copier-coller. Ensuite, vous appuierez sur la touche Entrée.

defaults write com.apple.dock pinning -string start; killall Dock

Si votre Dock est à gauche ou à droite, il démarrera du haut; s'il est en bas, il démarrera à gauche ! Amusant, hein ?

Petite remarque : pourquoi taper "killall Dock" ? Simplement pour "tuer" l'application Dock en cours afin qu'elle se relance initialisée avec le nouveau réglage. Maintenant, tapez :

defaults write com.apple.dock pinning -string end; killall Dock

Il démarre de l'autre côté ! Pour revenir à la normale, tapez :

defaults write com.apple.dock pinning -string middle; killall Dock

Voilà ! Terminé pour la petite leçon sur la commande Unix "defaults". Passons au script qui permet de se débarrasser d'un Dock "chiant".

Vous pouvez copier-coller tel quel tout ce qui se trouve sous les pointillés dans l'éditeur de scripts. Cliquez sur le bouton Compiler (pas sur Exécuter). Choisissez alors Enregistrer sous... dans le menu Fichier, et choisissez le format Application. Ne cochez rien. L'endroit approprié pour le stocker est le dossier Scripts de la bibliothèque de l'utilisateur (~ / Bibliothèque / Scripts /). Il sera alors disponible dans le menu Scripts de la barre des menus. (Ne le mettez pas dans le Dock, parce que si celui-ci devient microscopique... bonne chance).

Important : N'exécutez pas ce script à partir de l'éditeur de script, sous peine de perdre tous vos réglages (le Dock restera nain !). Les variables globales ne conservent leurs valeurs que si le script est sous forme d'application. Si cela vous arrive, ce n'est pas grave. Il suffit d'aller dans les Préférences système et d'y modifier les réglages du Dock pour le rendre à nouveau accessible.

-- L'objectif de ce script est de passer des réglages normaux du Dock à un Dock hypernain masqué placé sous la barre des menus, au milieu, et vice-versa. Quatre propriétés du Dock nous intéressent : autohide (masquage); magnification (agrandissement); orientation (position) et tileSize (taille).

-- La première chose à faire est de définir 4 variables globales qui conservent les réglages de l'utilisateur, même quand la machine est éteinte. Cela pour pouvoir rétablir le Dock dans ses réglages initiaux. Ces 4 variables seront initialisées à -- vide.

```
property monMasquage : ""  
property monAgrandissement : ""  
property maPosition : ""  
property maTaille : ""
```

-- Comme le script est réversible, le Dock sera normal une fois sur deux, et nain une fois sur deux également. Il est nécessaire de tester si le Dock est nanifié ou non. Comme le script place le Dock nain au sommet de l'écran, il suffit de lire la propriété "orientation" du Dock pour savoir dans quel état il se trouve : si la valeur est "top" (au sommet de l'écran), alors il est nain. Dans le cas contraire, il est normal. S'il est normal, on affecte alors aux 4 variables ci-dessus leurs valeurs actuelles.

```
set positionActuelle to do shell script "defaults read com.apple.dock orientation -string"  
if positionActuelle ≠ "top" then  
    set monMasquage to do shell script "defaults read com.apple.dock autohide -string"  
    set monAgrandissement to do shell script "defaults read com.apple.dock magnification -string"  
    set maPosition to positionActuelle  
    set maTaille to do shell script "defaults read com.apple.dock tileSize -number"  
end if
```

-- Remarquez que les propriétés "autohide" et "magnification" sont lues comme String et non comme Boolean, pour obtenir les chaînes "0" (pour "No") et "1" (pour "Yes").

-- On va maintenant afficher un dialogue de confirmation que l'utilisateur peut accepter ou non. Le dialogue sera différent
-- selon que le Dock est actuellement nanifié ou non.

```
if positionActuelle = "top" then
    display dialog "Réinstaller un Dock normal ?" buttons {"Non", "Oui"} default button 2 with icon 2
else
    display dialog "Camoufler le Dock ?" buttons {"Non", "Oui"} default button 2 with icon 2
end if
```

-- Le choix de l'utilisateur va être placé dans la variable leResultat

```
set leResultat to button returned of the result
```

-- On ne considère que le cas où l'utilisateur a cliqué sur "Oui".

```
if leResultat = "Oui" then
```

-- Au cas où le Dock est déjà miniaturisé (donc au sommet de l'écran), on restitue les valeurs stockées dans les variables
-- globales aux variables du Dock.

```
    if positionActuelle = "top" then
        if monMasquage = "1" then
            do shell script "defaults write com.apple.dock autohide -boolean Yes"
        else
            do shell script "defaults write com.apple.dock autohide -boolean No"
        end if
        if monAgrandissement = "1" then
            do shell script "defaults write com.apple.dock magnification -boolean Yes"
```

```
else
    do shell script "defaults write com.apple.dock magnification -boolean No"
end if
set scriptPosition to "defaults write com.apple.dock orientation -string " & maPosition
do shell script scriptPosition
set scriptTaille to "defaults write com.apple.dock tilesize -int " & maTaille
do shell script scriptTaille
do shell script "killall Dock"
```

-- Au cas où le Dock est normal, on lui attribue des valeurs qui vont le miniaturiser et le cacher au sommet de l'écran.

```
else
    do shell script "defaults write com.apple.dock autohide -boolean Yes"
    do shell script "defaults write com.apple.dock magnification -boolean No"
    do shell script "defaults write com.apple.dock orientation -string top"
    do shell script "defaults write com.apple.dock tilesize -int 2.000000"
    do shell script "killall Dock"
end if
end if
```

-- Pour la taille, la plus petite valeur est 2. Au dessous, le Dock reste miniaturisé sauf la corbeille !