

Close Message Window

Ferme la, ou les fenêtres, de message spécifiée et nettoye la mémoire de sa présence.

Syntaxe:

close message window *unObjet*

Paramètres:

unObjet

L'objet fenêtre de message, le nombre unique de la fenêtre de message, ou "all windows"

Classe: quelconque

Résultat:

Afficher l'heure, dans une nouvelle fenêtre, durant quelques secondes, puis fermer la fenêtre:

```
set lafenetreDeMessage to create message window at {10, 40} with properties {name:"Statut", closeable:false, message text:"L'heure:"}
```

```
set windowID to window ID of lafenetreDeMessage
```

```
delay 2
```

```
repeat with i from 1 to 10
```

```
  set ceTemps to afficheHeure(current date)
```

```
  update message window lafenetreDeMessage message text ceTemps
```

```
  delay 2
```

```
end repeat
```

```
close message window windowID
```

```
afficheHeure(current date)
```

```
on afficheHeure(laDate)
```

```
  copy word 5 of ((current date) as text) to laDate
```

```
  set exASdelimiters to AppleScript's text item delimiters
```

```
  set AppleScript's text item delimiters to {":"}
```

```
  set temp to number of text items in laDate
```

```
  set resultat to (text items 1 thru 3 of laDate) as text
```

```
  set AppleScript's text item delimiters to exASdelimiters
```

```
  return resultat
```

```
end afficheHeure
```

Close Progress Indicator

Ferme la, ou les fenêtres, de progression spécifiée et nettoye la mémoire de sa présence.

Syntaxe:

close progress indicator *unObjet*

Paramètres:

unObjet

L'objet indicateur de progression, le nombre unique de la fenêtre, ou "all windows"

Classe: quelconque

Résultat:

Afficher une nouvelle fenêtre de progression, durant quelques secondes, puis fermer la fenêtre:

```
property LeTitre : "Des routines s'exécutent..."
property LeCommentaire : "Attendre que la fenêtre se ferme."

set a to 0
try
  set x to create progress indicator with properties {top message:LeTitre, bottom
message:LeCommentaire, indeterminate:true, position:{100, 200}}
  repeat
    set a to a + 1
    delay 1
    if a = 10 then
      close progress indicator all windows
      quit
    end if
  end repeat
on error
end try

on quit
  tell me to quit
end quit
```

Create Message Window

Affiche un message texte dans une nouvelle fenêtre flottante.

Syntaxe:

```
create message window leTexte
  [ at unPoint ]
  [ with properties unEnregistrement ]
```

Paramètres:

leTexte

Le texte du message qu'il faut afficher.

Classe: chaîne de caractères ("string")

unPoint

La position de la fenêtre.

Classe: "point"

unEnregistrement

Les propriétés de la fenêtre (voir le détail de la classe "message window")

Classe: enregistrement ("record")

Résultat: Classe: "message window" -- les propriétés de la fenêtre de message.

Affichage d'une simple fenêtre:

set lafenetreDeMessage **to** create message window at {10, 400} -
with properties {name:"Simple Message", message text:"fenêtre construite avec 24U
Appearance OSAX.", closeable:true}

résultat: {window ID:1096064, position:{10, 400}, size:{350, 17}, bounds:{10, 400, 360,
417}, name:"Simple Message", closeable:true, resizable:false, kind:message window kind,
message text:"fenêtre construite avec 24U Appearance OSAX."}

Create Progress Indicator

Affiche une barre de progression dans une nouvelle fenêtre flottante.

Syntaxe:

```
create progress indicator  
  [ at unPoint ]  
  [ with properties unEnregistrement ]
```

Paramètres:

unPoint

La position de la fenêtre.

Classe: "point"

unEnregistrement

Les propriétés de la fenêtre (voir le détail de la classe "progress
indicator")

Classe: enregistrement ("record")

Résultat: Classe: "progress indicator" -- les propriétés de l'indicateur de
progression.

Affichage d'une simple fenêtre:

property LeTitre : "Des routines s'exécutent..."

property LeCommentaire : "Raccourci Commande-. pour fermer la fenêtre."

try

```
create progress indicator with properties {top message:LeTitre, bottom  
message:LeCommentaire, indeterminate:true, position:{100, 200}}
```

```
repeat
```

```
end repeat
```

```
on error
```

```
end try
```

Display Alert

Affiche une alerte standard grace à la fonction Appearance StandardAlert.

Syntaxe:

```
display alert leTexte  
  [ explanation leTexte2 ]  
  [ buttons lesBoutons ]  
  [ default button leBouton ]  
  [ cancel button leBouton2 ]  
  [ alert type unType ]
```

[movable *uneValeurBooleenne*]
[help button *uneValeurBooleenne2*]

Paramètres:

leTexte

Le texte principal qu'il faut afficher dans la boîte d'alerte.
Classe: chaîne de caractères ("string")

leTexte2

Le texte secondaire qu'il faut afficher en utilisant une petite police de caractère du système.
Classe: chaîne de caractères ("string")

lesBoutons

Une liste de noms (3 maximum) de boutons.
Classe: liste ("list")

leBouton

Le nom ou le numéro du bouton sélectionné par défaut.
Classe: chaîne de caractères ("string") ou nombre ("number")
Valeur par défaut:

leBouton2

Le nom ou le numéro du bouton d'annulation.
Classe: chaîne de caractères ("string") ou nombre ("number")

unType

Le type de boîte d'alerte qu'il faut afficher.
Classe: constante ("constant")
Valeur: stop/note/caution
Valeur par défaut: plain

uneValeurBooleenne

Est-ce que l'alerte se trouve dans une fenêtre déplaçable ?
Classe: valeur booléenne ("boolean")

uneValeurBooleenne2

Est-ce que le bouton d'aide doit être affiché ?
Classe: valeur booléenne ("boolean")

Résultat: Classe: "small integer" -- le numéro du bouton choisi.

Affichage de fenêtre avec commentaire, bouton d'aide et icône de note:

display alert -

"Exemple d'Appearance OSAX" explanation -

"Notez le bouton d'aide. " alert type note buttons {"OK"}**with** help button

Affichage de fenêtre déplaçable et icône de note:

display alert "Alerte pouvant être déplacée !" alert type caution buttons {"OK"} **with** movable

Affichage de fenêtre avec boutons:

display alert ↵

"Jusqu'à trois boutons..." explanation ↵

"Le bouton d'annulation, s'il existe, répond aux raccourcis clavier" & ↵

"Esc et Commande-!" alert type stop buttons {"OK", "Annuler", "Autre"} ↵

default button 1 cancel button 2

Display Better Dialog

Affiche une boîte de dialogue avec des champs texte, des champs cryptés, des cases à cocher, et autres...

Syntaxe:

display better dialog *leTexte*

[fields *lesChamps*]

[boutons *lesBoutons*]

[default button *leBouton*]

[cancel button *leBouton2*]

Paramètres:

leTexte

Le texte d'avertissement qu'il faut afficher dans la boîte de dialogue.

Classe: chaîne de caractères ("string")

lesChamps

Une liste (pouvant aller jusqu'à 10) de champs.

Classe: liste ("list")

lesBoutons

Une liste (pouvant aller jusqu'à 5) de noms de boutons.

Classe: liste ("list")

leBouton

Le nom ou le numéro de la case de commande sélectionnée par défaut.

Classe: chaîne de caractères ("string")

leBouton2

Le nom ou le numéro de la case de commande d'annulation.

Classe: chaîne de caractères ("string")

Résultat: Classe: "better dialog reply" -- un enregistrement contenant le bouton cliqué et les valeurs entrées (si elles existent).

Une fenêtre avec plusieurs champs:

set leResultat **to** display better dialog ↵

"Plusieurs champs possibles" fields {{kind:text field, name:"Votre nom:", value:"Gertrude"}, ↵

{kind:password field, name:"Votre mot-de-passe:", value:"zaza"}} ↵

buttons {"OK", "Annuler"}

résultat: {button returned:"Annuler", choices returned:{"Gertrude", "zaza"}, fields returned:{{kind:text field, name:"Votre nom:", value:"Gertrude"}, {kind:password field,

```
name:"Votre mot-de-passe:", value:"zaza"}}}
```

Une fenêtre avec plusieurs boutons radio:

```
set leResultat to display better dialog ~
```

```
"Vous pouvez utiliser des boutons radio:" fields {{kind:radio button, name:"Premier choix"}, ~  
{kind:radio button, name:"Second choix"}, ~  
{kind:radio button, name:"Troisième choix"}} ~  
buttons {"OK", "Annuler"}
```

```
résultat: {button returned:"Annuler", choices returned:{"Second choix"}, fields  
returned:{{kind:radio button, name:"Premier choix", value:unchecked state, selected:false},  
{kind:radio button, name:"Second choix", value:checked state, selected:true}, {kind:radio button,  
name:"Troisième choix", value:unchecked state, selected:false}}}
```

Une fenêtre avec des cases à cocher:

```
set leResultat to display better dialog ~
```

```
"Les cases à cocher:" fields {{kind:check box, name:"Premier choix", selected:true}, ~  
{kind:check box, name:"Second choix", selected:false}, ~  
{kind:check box, name:"Troisième choix", selected:true}, ~  
{kind:check box, name:"Quatrième choix", value:mixed state}} ~  
buttons {"OK", "Annuler"}
```

```
résultat: {button returned:"OK", choices returned:{"Premier choix", "Troisième choix",  
"Quatrième choix"}, fields returned:{{kind:check box, name:"Premier choix", selected:true,  
value:checked state}, {kind:check box, name:"Second choix", selected:false, value:unchecked  
state}, {kind:check box, name:"Troisième choix", selected:true, value:checked state},  
{kind:check box, name:"Quatrième choix", value:mixed state, selected:true}}}
```

Une fenêtre avec des menus locaux:

```
set lesChoix to {"Bretagne", "Auvergne", "Midi"}
```

```
set leResultat to display better dialog ~
```

```
"La version 1.2 vous offre aussi des menus locaux:" fields ~  
{{kind:popup menu, name:"Vos choix:", menu items:lesChoix}, ~  
{kind:popup menu, name:"Villes:", menu items:{"Nantes", "Plouay", "Murat", "Toulon"}}} ~  
buttons {"OK", "Annuler"}
```

```
résultat: {button returned:"OK", choices returned:{"Bretagne", "Nantes"}, fields  
returned:{{kind:popup menu, name:"Vos choix:", menu items:{"Bretagne", "Auvergne", "Midi"},  
value:"Bretagne"}, {kind:popup menu, name:"Villes:", menu items:{"Nantes", "Plouay",  
"Murat", "Toulon"}, value:"Nantes"}}}
```

... et maintenant mélangeons le tout:

```
set leResultat to display better dialog ~
```

```
"Fenêtre avec éléments hétérogènes:" fields ~  
{{kind:popup menu, name:"Type de compte:", menu items:{"POP3", "IMAP"}, value:2}, ~  
{kind:text field, name:"Serveur de courriel:", value:"mail.mac.com"}, ~  
{kind:text field, name:"Utilisateur:", value:"Gwénaël"}, ~  
{kind:password field, name:"Mot-de-passe:", value:"zaza"}, ~  
{kind:check box, name:"Enregistrer le mot-de-passe", value:checked state}} ~  
buttons {"OK", "Annuler"}
```

```
résultat: {button returned:"OK", choices returned:{"IMAP", "mail.mac.com", "Gwénaël",
```

```
"zaza", "Enregistrer le mot-de-passe"}, fields returned:{{kind:popup menu, name:"Type de
compte:", menu items:{"POP3", "IMAP"}, value:"IMAP"}, {kind:text field, name:"Serveur de
courriel:", value:"mail.mac.com"}, {kind:text field, name:"Utilisateur:", value:"Gwénaël"},
{kind:password field, name:"Mot-de-passe:", value:"zaza"}, {kind:check box, name:"Enregistrer
le mot-de-passe", value:checked state, selected:true}}}
```

Handle Floating Window

Envoyé à un script pour supporter un évènement généré par une fenêtre flottante.

Syntaxe:

```
handle floating window unObjet
  [ event leType ]
  [ button name leTexte ]
```

Paramètres:

unObjet

La fenêtre qui a générée l'évènement.

Classe: fenêtre flottante ("floating window")

leType

le type d'évènement.

Classe: constante ("constant")

Valeur: window closed/button clicked

leTexte

Le nom du bouton cliqué (n'oubliez pas de to check event type)

Classe: chaîne de caractères ("string")

Résultat:

Une fenêtre modifiée en cours d'exécution avec la fonction Handle:

```
global aBar, gErreur, aValeur, aTemps, aExecution
```

```
on run
```

```
run script "1" --une manière d'activer l' OSAX sans avoir besoin de redémarrer l'ordinateur
```

```
set aExecution to true
```

```
set gErreur to false
```

```
set aValeur to 0
```

```
set aTemps to (current date) + 10
```

```
try
```

```
set aBar to create progress indicator with properties -
```

```
{name:"Appearance OSAX Demo", closeable:true, indeterminate:true, maximum:50, top
message:"Attendre...", bottom message:"Calcul du temps restant...", buttons:{"Pause", "Stop"}}
```

```
on error uneErreur number numeroErreur
```

```
set gErreur to true
```

```
if numeroErreur = -1708 then
```

```
display dialog "Désolé, une erreur est intervenu, probablement parcequ' Appearance OSAX
n'a pas été installé correctement." buttons {"Quitter"} default button 1
```

```
else
```

```
error uneErreur number numeroErreur
```

```
end if
```

```
quit
```

```
end try
```

end run

on idle

if aExecution **then**

set aTD **to** aTemps - (current date)

if aTD \leq 0 **then**

set aValeur **to** aValeur + 2

update progress indicator aBar current value aValeur **without** indeterminate

update progress indicator aBar top message "Démonstration d'Appearance OSAX..."

update progress indicator aBar bottom message "Temps restant calculé approximativement:
& ((50 - aValeur) as string) & " seconds"

set aTemps **to** (current date) + 2

end if

if (aValeur \geq 50) **then** quit

end if

return 1

end idle

on quit

try

close progress indicator all windows

on error

end try

activate

continue quit

end quit

on handle floating window laFenetre event unEvenement button name leBouton

if leBouton **is** "Stop" **or** unEvenement **is** window closed **then**

quit

else if leBouton **is** "Pause" **then**

update progress indicator aBar buttons {"Continuer", "Stop"} **with** indeterminate

set aExecution **to** false

else if leBouton **is** "Continuer" **then**

update progress indicator aBar buttons {"Pause", "Stop"} **without** indeterminate

set aExecution **to** true

end if

end handle floating window

Post Notification

Envoit une notification en utilisant le Gestionnaire de Notification.

Syntaxe:

post notification

[*leTexte*]

[diamond mark *uneValeurBooleenne*]

[showing icon *unelcone*]

[playing *unSon*]

Paramètres:

leTexte

Le texte qu'il faut afficher dans le dialogue de notification.

Classe: chaîne de caractères ("string")

uneValeurBooleenne

Faut-il placer a diamond-shaped mark dans le menu Application.

Classe: valeur booléenne ("boolean")

Valeur par défaut: false

unelcone

Le nombre unique ID ou le nom d'une ressource de petite icône ('SICN') qui doit clignoter dans la barre de menu (quelques icônes du système: -16385 = horloge, -3998 = disquette, -3995 = disque dur, -3993 = poubelle, 0 = big diamond, 3 = Macintosh)

Classe: quelconque

unSon

Le nombre unique ID ou le nom d'une ressource 'snd ' qui doit être joué, ou le "bip système" pour le son standard du système.

Classe: quelconque

Résultat: Classe: "notification" -- l'objet notification.

property leMessage : "Cet applet réclame votre attention 3 fois lorsqu'il fonctionne en tâche de fond. Aller vers le Finder, et attendez..."

property leMessage2 : "Exemple de notification avec Appearance OSAX"

property leMessage3 : "Merci, je me sens mieux au devant de l'écran."

property leMessage4 : "Je me sens si seul! Amenez-moi au devant de l'écran, svp."

property leMessage5 : "He, vous ! Etes-vous là ? Amenez-moi au devant de l'écran !"

property uneAlerte : "Une erreur est intervenue; Appearance OSAX n'est sans doute pas installé correctement"

global gNotification, glsActif, gmonNom, gCompte, gErreur

on run

run script "1" --un moyen d'activer l'OSAX sans avoir besoin de redémarrer l'ordinateur.

set gErreur **to** false

set gCompte **to** 1

set gmonNom **to** GetmonNom()

set glsActif **to** false

try

display alert leMessage2 explanation leMessage alert type stop buttons {"OK"} default button

1

on error uneErreur number numeroErreur

set gErreur **to** true

if numeroErreur = -1708 **then**

display dialog uneAlerte buttons {"Quit"} default button 1

else

error uneErreur number numeroErreur

end if

quit

end try

end run

on idle

tell application "Finder" **to set** jeSuisDevant **to** frontmost **of** process gmonNom

if glsActif **then**

if jeSuisDevant **then**

remove notification gNotification

set glsActif **to** false

if gCompte > 3 **then**

quit

else

display alert leMessage3 alert type caution buttons {"OK"} default button 1

end if

else

```

    return 1
end if
else
    if not jeSuisDevant then
        PostNextNotification()
        return 1
    end if
end if
return 15
end idle

```

```

on quit
    if not gErreur then
        if glsActif then
            remove notification gNotification
            set glsActif to false
        end if
    end if
    continue quit
end quit

```

```

on GetmonNom()
    set monChemin to path to me as string
    set ancienDelimiteur to AppleScript's text item delimiters
    set AppleScript's text item delimiters to {":."}
    set monNom to last text item of monChemin
    set AppleScript's text item delimiters to ancienDelimiteur
    return monNom
end GetmonNom

```

```

on PostNextNotification()
    if gCompte = 1 then
        set gNotification to post notification showing icon "script" playing sound "Bonus" with
diamond mark
    else if gCompte = 2 then
        set gNotification to post notification leMessage4 with diamond mark
    else if gCompte = 3 then
        set gNotification to post notification leMessage5 showing icon "script" playing sound "Mult"
with diamond mark
    end if
    set glsActif to (gCompte is in {1, 2, 3})
    set gCompte to gCompte + 1
end PostNextNotification

```

Remove Notification

Supprime une notification créée par une instruction "post notification".

Syntaxe:

remove notification *unObjet*

Paramètres:

unObjet

L'objet de notification.

Classe: "notification"

Résultat:

Voir le script de l'exemple de la commande "Post Notification".

Update Message Window

Raffraichit quelques propriétés de la fenêtre de message spécifiée.

Syntaxe:

```
update message window unObjet  
  [ position unPoint ]  
  [ size unPoint2 ]  
  [ message text leTexte ]
```

Paramètres:

unObjet

L'objet fenêtre de message ou le nombre unique ID.

Classe: quelconque

unPoint

La nouvelle position de la fenêtre.

Classe: "point"

unPoint2

La nouvelle taille de la fenêtre.

Classe: "point"

leTexte

le nouveau texte de message.

Classe: texte international ("international text")

Résultat: Classe: "message window" -- les propriétés de la fenêtre de message.

Raffraichir le message d'une fenêtre:

```
set lafenetreDeMessage to create message window at {10, 40} with properties {name:"Statut",  
closeable:true}
```

```
set X to update message window lafenetreDeMessage message text "Attendre que l'utilisateur  
ferme la fenêtre de message..."
```

X

résultat X: {window ID:8954496, position:{10, 40}, size:{350, 17}, bounds:{10, 40, 360, 57}, name:"Statut", closeable:true, resizable:false, kind:message window kind, message text:"Attendre que l'utilisateur ferme la fenêtre de message..."}

Update Progress Indicator

Raffraichit quelques propriétés de la fenêtre d'indicateur de progression spécifiée.

Syntaxe:

```
update progress indicator unObjet  
  [ position unPoint ]  
  [ current value uneValeur ]  
  [ minimum leMinimum ]  
  [ maximum leMaximum ]  
  [ indeterminate uneValeurBooleenne ]
```

[top message *leTexte*]
[bottom message *leTexte2*]
[boutons *lesBoutons*]

Paramètres:

unObjet

L'objet fenêtre d'indicateur de progression ou le nombre unique ID.
Classe: quelconque

unPoint

La nouvelle position de la fenêtre.
Classe: "point"

uneValeur

La nouvelle valeur courante de la barre de progression.
Classe: petit nombre entier ("small integer")

leMinimum

La nouvelle valeur minimum de la barre de progression.
Classe: petit nombre entier ("small integer")

leMaximum

La nouvelle valeur maximum de la barre de progression.
Classe: petit nombre entier ("small integer")

uneValeurBooleenne

Faut-il afficher the barber pole indicator?
Classe: valeur booléenne ("boolean")

leTexte

Le message principal affiché above de la barre de progression.
Classe: texte international ("international text")

leTexte2

Le message secondaire affiché sous la barre de progression.
Classe: texte international ("international text")

lesBoutons

Les boutons.
Classe: liste ("list") de listes

Résultat: Classe: "progress indicator" -- les propriétés de l'indicateur de progression.

Raffraichir les messages d'une fenêtre de progression; ces messages affichent les dossiers visités (en haut) et les noms de fichiers de type image GIFf et JPEG (en bas); ce script est de type droplet, c'est-à-dire que vous glissez, sur le bureau, un dossier quelconque sur l'icône de son application pour qu'il s'exécute immédiatement :

property lesTypesFichiers : {"GIFf", "JPEG"}

property uneAlerte : "Désolé, une erreur est intervenu, probablement parcequ' Appearance

OSAX n'a pas été installé correctement."

global aBarre, aFichier

on ouvreFenetre()

run script "1" *--une manière d'activer l' OSAX sans avoir besoin de redémarrer l'ordinateur*

try

set aBarre to create progress indicator with properties ↵

{name:"Visiter les fichiers...", closeable:false, indeterminate:true, maximum:50, top message:"Attendre...", bottom message:"Visite des fichiers..."}

on error uneErreur number numeroErreur

if numeroErreur = -1708 then

display dialog uneAlerte buttons {"Quitter"} default button 1

else

error uneErreur number numeroErreur

end if

quit

end try

end ouvreFenetre

on quit

try

close progress indicator all windows

on error

end try

activate

continue quit

end quit

on open (LesFichiers)

script visiterFichiers

property listeFichiers : {}

on open (leFichier)

copy leFichier as string to end of listeFichiers

affiche(leFichier)

return listeFichiers

end open

end script

try

ouvreFenetre()

→ voir commande "walk folders" de Jon's Commands

set LesFichiers to walk folders LesFichiers with script visiterFichiers only using files of type lesTypesFichiers

quit

on error msg

display dialog msg

quit

end try

end open

on affiche(leFichier)

tell application "Finder"

set leNom to name of leFichier

set leDossier to name of folder of leFichier

end tell

set aFichier to leNom

update progress indicator aBarre top message "Dossier..." & leDossier

update progress indicator aBarre bottom message "Fichier:" & leNom

end affiche