

## Attach To

Attacher des documents à un message.

Syntaxe:

```
attach to uneRéférence documents lesAlias  
[ spooling uneValeurBooléenne ]
```

Paramètres:

### ***uneRéférence***

Le message auquel attacher des documents.  
Classe: référence ("reference")

### ***lesAlias***

La liste des documents qu'il faut attacher.  
Classe: liste ("list") d'"alias"

### ***uneValeurBooléenne***

La valeur est mise à *true* pour faire qu'Eudora copie le document et attache la copie, puis détruit la copie lorsque le message est supprimé.  
Classe: valeur booléenne ("boolean")

Résultat:

```
set x to choose file -- voir "choose file"  
tell application "Eudora"  
  activate  
  make message attend of mailbox 2  
  tell message "" to attach to it documents {alias x} without spooling  
end tell
```

## Close

Fermer un objet.

Syntaxe:

```
close uneRéférence
```

Paramètres:

### ***uneRéférence***

L'objet qu'il faut fermer.  
Classe: référence ("reference")

Résultat:

```
tell application "Eudora"  
  activate  
  close window "address book"  
  close window "signatures"  
  close window "statistics"  
  close window "task progress"  
end tell
```

## Connect

Se connecter au serveur de courrier et transmettre le courrier.

Syntaxe:

connect

[ sending *uneValeurBooléenne* ]

[ checking *uneVérification* ]

[ waiting *uneAttente* ]

Paramètres:

### ***uneValeurBooléenne***

La valeur est mise à *true* pour faire qu'Eudora envoie les messages en file d'attente.

Classe: valeur booléenne ("boolean")

### ***uneVérification***

La valeur est mise à *true* pour faire qu'Eudora vérifie l'existence de message à importer.

Classe: valeur booléenne ("boolean")

### ***uneAttente***

La valeur est mise à *true* pour faire qu'Eudora attende un message "idle" avant toute opération de vérification.

Classe: valeur booléenne ("boolean")

Résultat:

```
tell application "Eudora"  
  activate  
  connectwith sending without checking and waiting  
end tell
```

## **Count**

Retourne le nombre d'éléments d'une classe donnée qui se trouvent dans un objet.

Syntaxe:

count *uneRéférence* each *laClasse*

Paramètres:

### ***uneRéférence***

L'objet dont les éléments doivent être comptés.

Classe: référence ("reference")

### ***laClasse***

La classe des éléments qui doivent être comptés. Le terme **each** est optionnel dans AppleScript.

Classe: classe ("class")

Résultat: Classe: nombre entier ("integer"). Le nombre d'éléments.

```
tell application "Eudora"  
  count messagesof mailbox 1  
end tell
```

résultat: 3

## **Delete**

Détruire un élément d'un objet.

Syntaxe:

delete *uneRéférence*

Paramètres:

***uneRéférence***

L'élément qu'il faut détruire.

Classe: référence ("reference")

Résultat:

```
tell application "Eudora"  
delete nickname "zaza"  
end tell
```

*La commande Delete n'est pas utilisée par Eudora, sauf pour détruire des personnalités ("personality") et des surnoms ("nickname"). Ce terme n'est donc pas utilisé pour la destruction de messages ("message"), de boîtes aux lettres ("mailbox"), etc... voir déplacer ("move").*

## Duplicate

Dupliquer un objet.

Syntaxe:

duplicate *uneRéférenceAobjet* to *uneRéférence*

Paramètres:

***uneRéférenceAobjet***

L'objet qu'il faut dupliquer.

Classe: référence ("reference")

***uneRéférence***

L'emplacement du nouvel objet.

Classe: référence ("reference")

Résultat: Classe: référence ("reference"). La référence au nouvel objet créé.

```
tell application "Eudora"  
set x to count messages of mailbox "out"  
duplicate message x of mailbox "out" to end of mailbox "test"  
end tell
```

*La commande Duplicate est utilisé avec l'expression "end of" pour le déplacement d'une boîte à lettre à une autre.*

## Exists

Vérifie si un objet existe.

Syntaxe:

exists *uneRéférence*

Paramètres:

***uneRéférence***

L'objet dont l'on veut s'assurer de l'existence.

Classe: référence ("reference")

Résultat: Classe: valeur booléenne ("boolean"). *true* (vrai) si l'objet existe, sinon *false* (faux).

```
tell application "Eudora"  
  set x to count messages of mailbox "out"  
  exists message (x + 1)of mailbox "out"  
end tell
```

résultat: false

## Fetchurl

Retourne le contenu d'un URL.

Syntaxe:

```
fetchurl leTexte
```

Paramètres:

***leTexte***

L'URL que l'on va visiter (finger/ph/mailto).

Classe: chaîne de caractères ("string")

Résultat: Classe: texte international ("international text"). Le contenu de l'URL.

## Forward

Faire suivre un message.

Syntaxe:

```
forward uneRéférence
```

Paramètres:

***uneRéférence***

Le message que l'on fait suivre.

Classe: référence ("reference")

Résultat: Classe: référence ("reference"). La référence au message que l'on fait suivre.

## Get

Récupère les données d'un objet.

Syntaxe:

```
get uneRéférence
```

Paramètres:

***uneRéférence***

L'objet dont les données seront retournées.

Classe: référence ("reference")

Résultat: Classe: quelconque. Les données de l'objet.

```
tell application "Eudora"  
  get count messages of mailbox 1
```

**end tell**

résultat: 11

## Geturl

Affiche le contenu d'un URL.

Syntaxe:

**geturl** *leTexte*

Paramètres:

***leTexte***

L'URL que l'on se propose d'afficher (finger/ph/mailto).

Classe: chaîne de caractères ("string")

Résultat:

```
tell application "Eudora"
```

```
geturl
```

```
"file:///gwen/scripting/Frontier%205.0.1/Websites/Cormoran/as/osa/appl/frontier/dico/b  
oucle/loop.html"
```

```
end tell
```

## Make

Créer un nouvel élément.

Syntaxe:

**make new** *laClasse* at *uneRéférence*

[ invisible *valeurBooleenne* ]

[ with data *quelconque* ]

Paramètres:

***laClasse***

La classe du nouvel élément. Le terme 'new' est optionnel avec AppleScript.

Classe: type classe ("class")

***uneRéférence***

L'emplacement où insérer le nouvel élément.

Classe: référence ("reference")

***valeurBooleenne***

La valeur est *true* pour rendre invisible le nouveau message.

Classe: valeur booléenne ("boolean")

***quelconque***

Les données initiales de l'élément.

Classe: quelconque

Résultat: Classe: référence ("reference"). La référence au nouvel objet.

```
tell application "Eudora"
```

```
activate --voir "activate"
```

```
set leNom to expansion of nickname "gwen" -- voir "nickname (prop)"
```

```
make message at end of mailbox "out" of mail folder "" -- voir "mailbox", "mail folder"
```

**set** field "to" **of** message 0 **to** leNom  
**end tell**

## Move

Déplacer un objet à un nouvel emplacement.

Syntaxe:

*move* *uneRéférenceAObjet* **to** *uneRéférence*

Paramètres:

***uneRéférenceAObjet***

L'objet qu'il faut déplacer.

Classe: référence ("reference")

***uneRéférence***

Le nouvel emplacement de L'objet.

Classe: référence ("reference")

Résultat: Classe: référence ("reference"). La référence à un objet déplacé, après son déplacement.

**tell** application "Eudora"

*activate* --voir "*activate*"

*move* message 1 **of** mailbox "out" **to end of** mailbox "test" -- voir "*message*", "*mailbox*"

**end tell**

résultat: message id 2.102820289E+9 of mailbox "test" of application "Eudora"

Mettre une lettre à la poubelle.

**tell** application "Eudora"

*activate* --voir *Activate*

*move* message 1 **of** mailbox "in" **to end of** mailbox "trash"

**end tell**

résultat: message id 1.812717085E+9 of mailbox "trash" of application "Eudora"

*La commande Move est utilisé avec l'expression "end of" pour le déplacement d'une boîte à lettre à une autre.*

## Notice

Eudora envoie cet évènement pour avertir une application que quelque chose s'est passé.

Syntaxe:

*notice*

[ occurrence *unEvenement* ]

[ messages *uneListe* ]

Paramètres:

***unEvenement***

Quel évènement est intervenu ?

Classe: constante ("constant")

Valeur: *mail arrives* / *mail sent* / *will connect* / *has connected* / *has manually filtered*

***uneListe***

Les messages.  
Classe: liste ("list")

Résultat:

## Open

Ouvrir un objet spécifié.

Syntaxe:

open *uneRéférence*

Paramètres:

***uneRéférence***

Une liste d'objets à ouvrir.

Classe: référence ("reference") ou liste ("list") de références

Résultat:

Ouvrir un fichier texte.

```
set x to choose file -- voir "choose file"  
tell application "Eudora"  
  activate  
  open {alias x}  
end tell
```

Ouvrir une fenêtre.

```
tell application "Eudora"  
  activate  
  open window "address book"  
  open window "signatures"  
  open window "statistics"  
  open window "task progress"  
end tell
```

Ouvrir une boîte à lettres.

```
tell application "Eudora"  
  activate  
  set {entree, sortie} to {file of mailbox "in", file of mailbox "out"}  
  open {entree, sortie}  
end tell
```

## Print

Imprimer le message spécifié.

Syntaxe:

print *uneRéférence*

Paramètres:

***uneRéférence***

Le message à imprimer.

Classe: référence ("reference")

Résultat:

```
tell application "Eudora"  
  print message 1 of mailbox 1  
end tell
```

## Queue

Met un message dans la liste des messages prêts à être envoyés (dans la boîte à lettre ("mailbox") "out").

Syntaxe:

```
queue uneRéférence  
  [ for laDate ]
```

Paramètres:

### ***uneRéférence***

Le message à mettre en file d'attente.

Classe: référence ("reference")

### ***laDate***

La date et l'heure pour envoyer le message.

Classe: "date"

Résultat:

```
tell application "Eudora"  
  activate  
  set x to make message at (end of mailbox "Out" of mail folder "")  
  set field "To" of x to "LeNom@domaine.com"  
  set field "Subject" of x to "Le Sujet"  
  set field "" of x to "corps du texte"  
  queue x  
end tell
```

## Quit

Quitte une application.

Syntaxe:

```
quit
```

Paramètres:

Résultat:

```
tell application "Eudora"  
  quit  
end tell
```

## Redirect

Redirige un message.

Syntaxe:

```
redirect uneRéférence
```

Paramètres:

### ***uneRéférence***

Le message à rediriger.

Classe: référence ("reference")

Résultat: Classe: référence ("reference"). La référence à un message redirigé.

```
tell application "Eudora"
activate
set x to make message at (end of mailbox "Out" of mail folder "")
set field "To" of x to "LeNom@domaine.com"
set field "Subject" of x to "Le Sujet"
set field "" of x to "corps du texte"
set leMessage to redirect message 0
tell leMessage
  set field "To" to "LeNom2@domaine2.com"
  set field "Reply-to" to "MonNom@MonDomaine.com"
  set field "Subject" to "Le Sujet"
queue
end tell
queue x
end tell
```

## Reply

Répondre à un message. Cette commande crée, par défaut, une réponse au message de l'envoyeur, qui n'est pas obligatoirement l'auteur, ou à l'adresse indiquée dans le champ **Reply-To:** , si ce champ existe.

### Syntaxe:

```
reply uneRéférence
[ quoting desGuillemets ]
[ everyone toutLeMonde ]
[ self moiMeme ]
```

### Paramètres:

#### ***uneRéférence***

Le message auquel répondre.  
Classe: référence ("reference")

#### ***desGuillemets***

La valeur doit être mise à *true* (vrai) si vous voulez que le texte original apparaisse entre guillemets dans la réponse. Par défaut, la valeur est *true* (vrai).

Classe: valeur booléenne ("boolean")

#### ***toutLeMonde***

La valeur doit être mise à *true* si vous voulez que la réponse doit parvenir à toute personne ayant reçu le message original.

Classe: valeur booléenne ("boolean")

#### ***moiMeme***

La valeur doit être mise à *true* si vous voulez que la réponse vous parvienne aussi.

Classe: valeur booléenne ("boolean")

Résultat: Classe: référence ("reference"). La référence au message de réponse.

```
tell application "Eudora"  
activate  
replythe front message with everyone without quoting  
end tell
```

résultat: message id 2.123204284E+9 of mailbox "Out" of mail folder "" of application "Eudora"

## Run

Cette commande est envoyée à une application lorsque son icône est double-cliquée.

Syntaxe:

```
run
```

Paramètres:

Résultat:

```
tell application "Eudora"  
run  
end tell
```

## Save

Sauve, c'est-à-dire enregistre sur disque dur, un objet.

Syntaxe:

```
save uneRéférence
```

Paramètres:

***uneRéférence***

Le message à sauver.

Classe: référence ("reference")

Résultat:

```
tell application "Eudora"  
make message atend of mailbox "out"  
set field "to" of message 0 to "nom@zaza.com"  
set field "subject" of message 0 to "le sujet"  
set field "" of message 0 to "le corps du texte" -- ou set body of message 0 to ...  
save message 0  
end tell
```

## Send Again

Envoyer à nouveau un message.

Syntaxe:

```
send again uneRéférence
```

Paramètres:

***uneRéférence***

Le message qu'il faut envoyer à nouveau.

Classe: référence ("reference")

Résultat:

Classe: référence ("reference"). Une référence au message qu'il faut envoyer à nouveau.

```
tell application "Eudora"  
tell (send again message 1 of mailbox "In" of mail folder "")  
  set field "subject" to "a voir"  
  queue  
end tell  
end tell
```

## Set

Affecter des données à un objet.

Syntaxe:

**set** *uneRéférence* **to** *quelconque*

Paramètres:

### ***uneRéférence***

L'objet qu'il faut modifier.

Classe: référence ("reference")

### ***quelconque***

La nouvelle valeur.

Classe: quelconque

Résultat:

```
tell application "Eudora"  
  make message attend of mailbox "out"  
  set field "to" of message 0 to "nom@zaza.com"  
end tell
```

## Start Notifying

Avertir une application des évènements qui se déroulent.

Syntaxe:

```
start notifying uneRéférence  
  [ when unEvenement ]
```

Paramètres:

### ***uneRéférence***

Une application qu'il faut avertir d'un évènement.

Classe: référence ("reference")

### ***unEvenement***

De quel évènement déroulé faut-il avertir l'application ?

Classe: constante ("constant")

Valeur: *mail arrives / mail sent / will connect / has connected / has manually filtered*

Résultat:

## Stop Notifying

Arrêter d'avertir les applications des évènements qui se sont déroulés.

Syntaxe:

stop notifying

[ *uneRéférence* ]

[ when *unEvenement* ]

Paramètres:

***uneRéférence***

Une application qui est actuellement avertie des évènements qui se déroulent. Si ce paramètre est omis, alors toutes les applications.

Classe: référence ("reference")

***unEvenement***

Les évènements dont l'on ne désire plus qu'ils soient signifiés à l'application.

Classe: constante ("constant")

Valeur: *mail arrives / mail sent / will connect / has connected / has manually filtered*

Résultat:

## Unqueue

Supprimer un message de la file d'attente des messages prêts pour une expédition. Ce faisant, il ne sera pas envoyé.

Syntaxe:

unqueue *uneRéférence*

Paramètres:

***uneRéférence***

Le message à supprimer de la file d'attente.

Classe: référence ("reference")

Résultat: