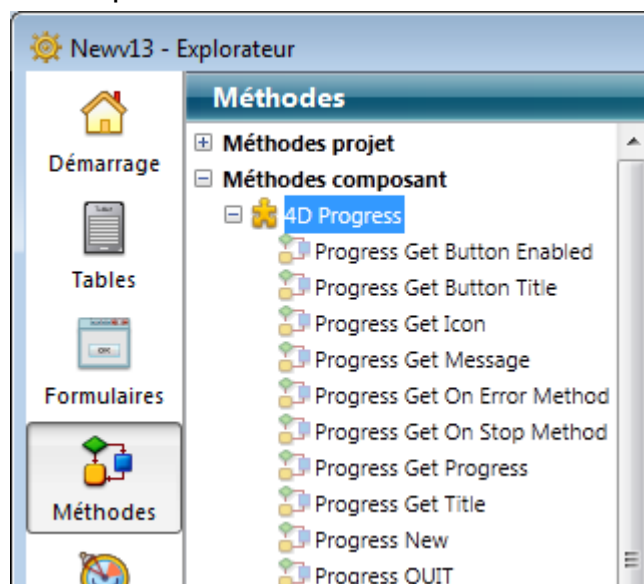


4D Progress

4D contient un composant intégré nommé 4D Progress. Ce composant vous permettra d'ouvrir une ou plusieurs barres de progression dans une même fenêtre (à l'image de l'interface du Finder de Mac OS).

Chaque barre de progression se voit attribuer un ID, généré automatiquement par la méthode **Progress New**, qui sera utilisé par toutes les méthodes projet du composant pour gérer les propriétés et les valeurs dans les boîtes de dialogue de progression.

Les méthodes projet du composant sont listées dans la page **Méthodes composant** de l'Explorateur :



Barres de progression



Liste alphabétique des commandes



Barres de progression

- ⚙ Progress Get Button Enabled
- ⚙ Progress Get Button Title
- ⚙ Progress Get Icon
- ⚙ Progress Get Message
- ⚙ Progress Get On Error Method
- ⚙ Progress Get On Stop Method
- ⚙ Progress Get Progress
- ⚙ Progress Get Title
- ⚙ Progress New
- ⚙ Progress QUIT
- ⚙ Progress SET BUTTON ENABLED
- ⚙ Progress SET BUTTON TITLE
- ⚙ Progress SET FONT SIZES
- ⚙ Progress SET FONTS
- ⚙ Progress SET ICON
- ⚙ Progress SET MESSAGE
- ⚙ Progress SET ON ERROR METHOD
- ⚙ Progress SET ON STOP METHOD
- ⚙ Progress SET PROGRESS
- ⚙ Progress SET TITLE
- ⚙ Progress SET WINDOW VISIBLE
- ⚙ Progress Stopped

Progress Get Button Enabled

Progress Get Button Enabled (id) -> Résultat

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Booléen	↩	Vrai = bouton Stop affiché, sinon Faux

Description

La méthode **Progress Get Button Enabled** retourne **Vrai** si la barre de progression désignée par le paramètre *id* affiche un bouton **Stop**. Si la barre n'affiche pas de bouton **Stop** (fonctionnement par défaut), la méthode retourne **Faux**.

Progress Get Button Title

Progress Get Button Title (id) -> Résultat

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Texte	↩	Libellé du bouton Stop

Description

Note: Cette méthode est utilisable sous Windows uniquement. Sous Mac OS, le bouton d'arrêt n'a pas de libellé.

La méthode **Progress Get Button Title** retourne le libellé courant du bouton d'arrêt de la barre de progression désignée par le paramètre *id*.

Par défaut, le libellé est "Stop". A noter que la méthode retourne le libellé courant même si le bouton **Stop** n'est pas affiché.

Progress Get Icon

Progress Get Icon (id) -> Résultat

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Image	↪	Icône de la barre de progression

Description

La méthode **Progress Get Icon** retourne l'icône de la barre de progression désignée par le paramètre *id*.

Progress Get Message

Progress Get Message (id) -> Résultat


Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Texte	↪	Message de la barre de progression

Description

La méthode **Progress Get Message** retourne le message de la barre de progression désignée par le paramètre *id*.

Progress Get On Error Method

Progress Get On Error Method -> Résultat

Paramètre	Type	Description
Résultat	Texte	 Nom de la méthode appelée en cas d'erreur (si définie)

Description

La méthode **Progress Get On Error Method** retourne le nom de la méthode projet de la base hôte appelée en cas d'erreur lors de l'utilisation d'une barre de progression.

Si aucune méthode d'erreur n'est définie, la méthode retourne une chaîne vide.

Progress Get On Stop Method

Progress Get On Stop Method (id) -> Résultat

Paramètre	Type	Description
id	Entier long	➡ ID de la barre de progression
Résultat	Texte	➡ Nom de la méthode appelée en cas de clic sur le bouton Stop (si définie)

Description

La méthode **Progress Get On Stop Method** retourne le nom de la méthode projet de la base hôte appelée lorsque l'utilisateur clique sur le bouton **Stop** de la barre de progression désignée par le paramètre *id*.

Si aucune méthode n'est associée au **Stop**, la méthode retourne une chaîne vide.

Progress Get Progress

Progress Get Progress (id) -> Résultat

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Réel	↩	Valeur de la barre de progression

Description

La méthode **Progress Get Progress** retourne la valeur courante associée à la barre de progression désignée par le paramètre *id*.

Progress Get Title

Progress Get Title (id) -> Résultat

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
Résultat	Texte	↩	Titre de la barre de progression

Description

La méthode **Progress Get Title** retourne le titre principal de la barre de progression désignée par le paramètre *id*.

Progress New

Progress New -> Résultat

Paramètre	Type	Description
Résultat	Entier long	 ID de la nouvelle barre de progression

Description

La méthode **Progress New** crée une nouvelle barre de progression et retourne son numéro d'ID. Ce numéro est unique pendant la durée de vie de la barre de progression mais pourra être réutilisé.

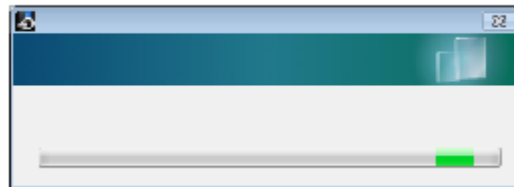
Au premier appel de cette méthode, un process local est créé et une nouvelle fenêtre centrée est ouverte au-dessus de la fenêtre principale. Par défaut, cette fenêtre :

- contient une barre de progression indéfinie
- n'a pas de message.

Mac OS



Windows



Progress QUIT

Progress QUIT (id)

Paramètre	Type	Description
id	Entier long →	ID de la barre de progression (0 = toutes les barres)

Description

La méthode **Progress QUIT** vous permet de fermer la barre de progression référencée par le paramètre *id*.

Si *id* désigne la seule barre de progression affichée, la fenêtre de progression est également refermée (ainsi que le process local). Sinon, la fenêtre est redimensionnée afin de ne contenir que les barres ouvertes.

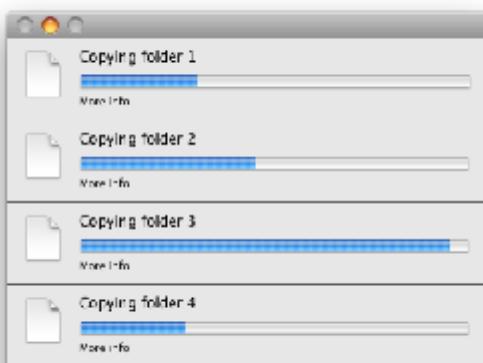
Vous pouvez passer 0 dans le paramètre *id* afin de stopper toutes les barres de progression et de refermer la fenêtre de progression.

Exemple

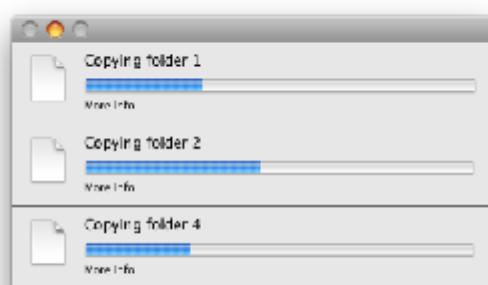
Si la barre nommée "Copying folder 3" a le numéro d'ID 3 :

Progress QUIT(3)

Avant

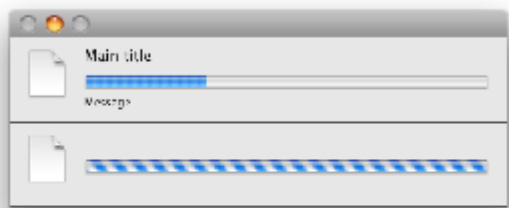


Après

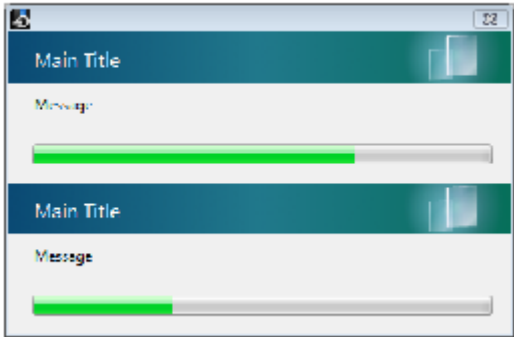


Si une fenêtre de progression était déjà ouverte à l'appel de cette méthode, elle est redimensionnée afin d'afficher une nouvelle barre de progression sous la ou les précédente(s), dans le même process :

Mac OS



Windows



Progress SET BUTTON ENABLED

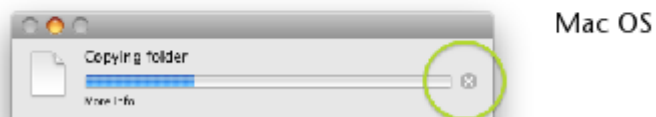
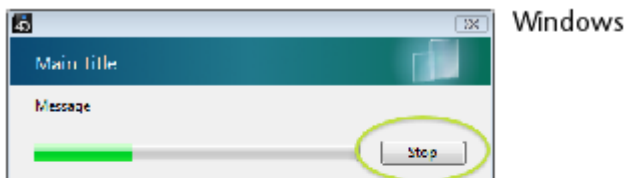
Progress SET BUTTON ENABLED (id ; bouton)

Paramètre	Type		Description
id	Entier long	➡	ID de la barre de progression
bouton	Booléen	➡	Vrai = Afficher, Faux = Masquer

Description

La méthode **Progress SET BUTTON ENABLED** vous permet d'ajouter un bouton **Stop** à la barre de progression désignée par le paramètre *id*.

Par défaut, les barres de progression n'ont pas de bouton **Stop**. Si vous passez **Vrai** dans le paramètre bouton, un bouton sera affiché :



Vous pouvez gérer l'effet du clic sur le bouton **Stop** à l'aide de la méthode **Progress SET ON STOP METHOD** ou en testant la valeur de la méthode **Progress Stopped**.

Progress SET BUTTON TITLE

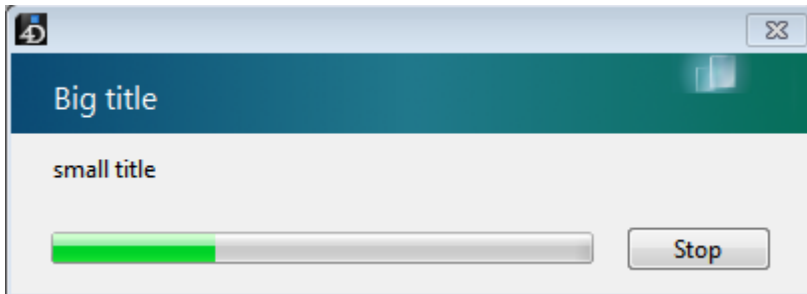
Progress SET BUTTON TITLE (id ; titre)

Paramètre	Type		Description
id	Entier long	➡	ID de la barre de progression
titre	Texte	➡	Libellé du bouton Stop (Windows)

Description

Note: Cette méthode est utilisable sous Windows uniquement. Sous Mac OS, le bouton d'arrêt n'a pas de libellé.

La méthode **Progress SET BUTTON TITLE** vous permet de modifier le libellé du bouton **Stop** de la barre de progression désignée par le paramètre *id*. Par défaut, le libellé du bouton d'arrêt est "Stop" :

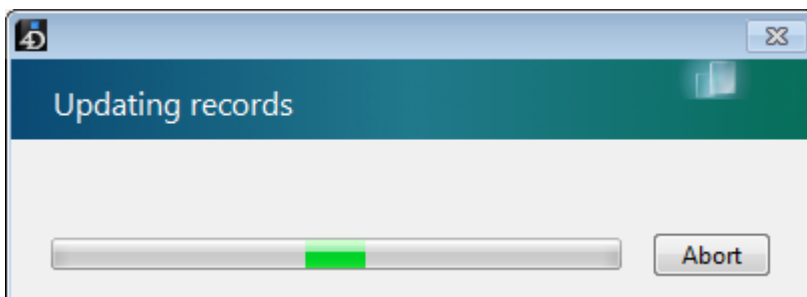


Par défaut, les barres de progression n'ont pas de bouton **Stop**. Pour qu'une barre de progression contienne un bouton **Stop**, vous devez utiliser la méthode **Progress SET BUTTON ENABLED**.

Exemple

Vous souhaitez utiliser le libellé "Abort" :

```
<>ID:=Progress New
...
Progress SET BUTTON TITLE(<>ID;"Abort")
```



Progress SET FONT SIZES

Progress SET FONT SIZES (tailleTitres {; tailleMessages {; tailleBoutons}})

Paramètre	Type	Description
tailleTitres	Entier long	→ Taille de police pour les titres
tailleMessages	Entier long	→ Taille de police pour les messages
tailleBoutons	Entier long	→ (Windows) Taille de police pour les boutons Stop

Description

La méthode **Progress SET FONT SIZES** vous permet de modifier la taille des polices de caractères utilisées pour les différents textes affichés dans toutes les fenêtres de progression.

Passez dans les paramètres *tailleTitres*, *tailleMessages* et *tailleBoutons* les tailles des polices de caractères à utiliser. Si vous ne souhaitez pas modifier une taille, passez -1 dans le paramètre correspondant.

Exemple 1

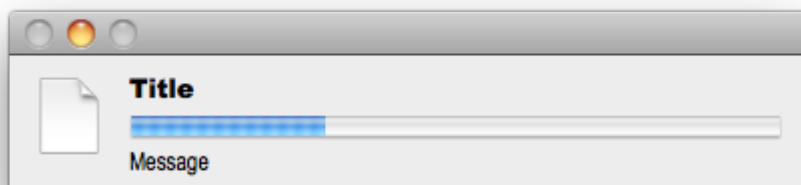
On souhaite changer uniquement la taille des messages :

```
Progress SET FONT SIZES(-1;13)
```

Exemple 2

On change la police et la taille des titres et des messages :

```
Progress SET FONTS("Arial Black";"Arial narrow")  
Progress SET FONT SIZES(13;12)
```



Progress SET FONTS

Progress SET FONTS (policeTitres {; policeMessages {; policeBoutons}})

Paramètre	Type	Description
policeTitres	Texte →	Police à utiliser pour les titres
policeMessages	Texte →	Police à utiliser pour les messages
policeBoutons	Texte →	(Windows) Police à utiliser pour les boutons Stop

Description

La méthode **Progress SET FONTS** vous permet de modifier les polices de caractères utilisées pour les différents textes affichés dans toutes les fenêtres de progression.

Passez dans les paramètres *policeTitres*, *policeMessages* et *policeBoutons* les noms des polices de caractères à utiliser. Si vous ne souhaitez pas modifier une police, passez une chaîne vide ("") dans le paramètre correspondant.

Exemple

On souhaite changer uniquement la police des messages :

```
Progress SET FONTS("";"Arial")
```

Progress SET ICON

Progress SET ICON (id ; icône { ; premierPlan })

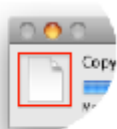
Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
icône	Image	→	Image à afficher en tant qu'icône
premierPlan	Booléen	→	Montrer la progression au premier plan

Description

La méthode **Progress SET ICON** vous permet de modifier l'icône affichée dans la barre de progression. Par défaut, les icônes suivantes sont affichées :



Windows



Mac OS

Passez dans *id* le numéro d'ID unique de la barre de progression, retourné par la méthode **Progress New**.

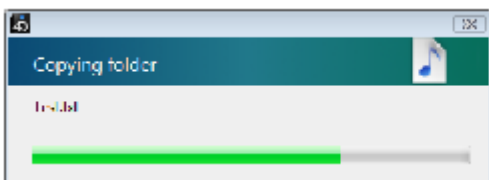
Passez dans *icône* l'image (variable ou champ) à utiliser comme icône dans la fenêtre de la barre de progression. La taille maximale de cette icône doit être :

- sous Mac OS, 40 x 40 pixels
- sous Windows, 40 x 80 pixels

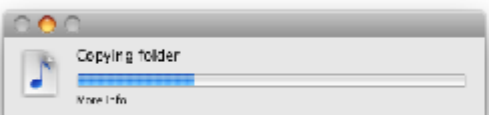
Si vous passez une icône de taille inférieure à ces limites, elle sera centrée et non redimensionnée. Si elle est de taille supérieure, elle sera centrée et redimensionnée.

Passez **Vrai** dans *premierPlan* si vous voulez forcer le passage de la fenêtre de progression au premier plan de l'application.

Exemples d'icônes personnalisées :



Windows



Mac OS

Progress SET MESSAGE

Progress SET MESSAGE (id ; message {; premierPlan})

Paramètre	Type		Description
id	Entier long	➡	ID de la barre de progression
message	Texte	➡	Message de la barre
premierPlan	Booléen	➡	Montrer la progression au premier plan

Description

La méthode **Progress SET MESSAGE** vous permet de modifier le message affiché dans la barre de progression.

Passez dans *id* le numéro d'ID unique de la barre de progression, retourné par la méthode **Progress New**.

Passez dans *message* le texte à modifier sous le titre principal (Windows) ou sous la barre de progression (Mac OS).

Passez **Vrai** dans *premierPlan* si vous voulez forcer le passage de la fenêtre de progression au premier plan de l'application.

Progress SET ON ERROR METHOD

Progress SET ON ERROR METHOD (*nomMéthode*)

Paramètre	Type		Description
<i>nomMéthode</i>	Texte	→	Nom de la méthode d'erreur

Description

La méthode **Progress SET ON ERROR METHOD** vous permet de désigner une méthode à exécuter en cas d'erreur lors de l'utilisation des barres de progression (par exemple id inconnu, nombre de paramètres incorrect, etc.).

Passez dans *nomMéthode* le nom de la méthode projet de la base hôte à appeler en cas d'erreur. Cette méthode sera commune à toutes les fenêtres de progression de l'application.

Note : Attention, comme la méthode **Progress SET ON ERROR METHOD** appartient à un composant, vous devez veiller à affecter la propriété "Partagée entre composants et base hôte" à la méthode *nomMéthode*, sinon une erreur sera retournée.

La méthode *nomMéthode* sera appelée avec trois paramètres :

- \$1 (Entier long) : numéro de l'erreur
- \$2 (Texte) : texte de l'erreur
- \$3 (Entier long) : ID unique de la barre de progression

Exemple

Voici un exemple de méthode appelée en cas d'erreur. Cette méthode a été déclarée "partagée" en mode Développement :

```
C_ENTIER LONG($1)
C_TEXTE($2)
C_ENTIER LONG($3)

C_ENTIER LONG($ErrorID)
C_TEXTE($ErrorText)
C_ENTIER LONG($ProgressID)

$ErrorID:=$1
$ErrorText:=$2
$ProgressID:=$3

$Error:=""
$Error:=$Error+"Erreur numéro : "+Chaine($ErrorID)+Caractere(Retour chariot)
$Error:=$Error+$ErrorText+Caractere(Retour chariot)
```

```
$Error:=$Error+"ID progression : "+Chaine($ProgressID)  
ALERTE($Error)
```

Progress SET ON STOP METHOD

Progress SET ON STOP METHOD (id ; nomMéthode)

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
nomMéthode	Texte	→	Nom de la méthode de stop

Description

La méthode **Progress SET ON STOP METHOD** vous permet de désigner une méthode à exécuter lorsque l'utilisateur clique sur le bouton **Stop** de la barre de progression.

Par défaut, une barre de progression ne contient pas de bouton **Stop**. Vous devez l'afficher explicitement à l'aide de la méthode **Progress SET BUTTON ENABLED**.

Passez dans *id* le numéro d'ID unique de la barre de progression, retourné par la méthode **Progress New**.

Passez dans *nomMéthode* le nom de la méthode projet de la base hôte à appeler lors du clic sur le bouton **Stop**. Cette méthode recevra l'ID unique de la barre de progression en premier paramètre. Elle sera exécutée dans un nouveau process lancé par le composant.

Note: Attention, comme la méthode **Progress SET ON STOP METHOD** appartient à un composant, vous devez veiller à affecter la propriété "Partagée entre composants et base hôte" à la méthode *nomMéthode*, sinon une erreur sera retournée.

Progress SET PROGRESS

Progress SET PROGRESS (id ; progression {; message {; premierPlan}})

Paramètre	Type		Description
id	Entier long	➡	ID de la barre de progression
progression	Réel	➡	Valeur de progression ([0...1] ou -1)
message	Texte	➡	Message de la barre
premierPlan	Booléen	➡	Montrer la progression au premier plan

Description

La méthode **Progress SET PROGRESS** vous permet de modifier la valeur de la barre de progression ainsi que les informations affichées dans la fenêtre de progression. Cette méthode est utile pour la mise à jour d'une barre de progression au sein d'une boucle.

Passez dans *id* le numéro d'ID unique de la barre de progression, retourné par la méthode **Progress New**.

Passez dans *progression* la valeur courante de la barre de progression. Vous pouvez passer une valeur réelle (entre 0 et 1) ou -1 afin de définir une barre de progression indéfinie (aussi appelée "Barber shop" sous Mac OS).

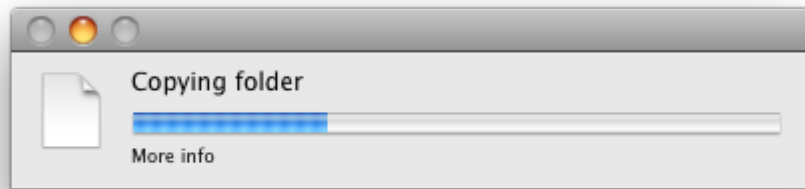
Passez dans *message* un texte additionnel à afficher sous le titre principal (Windows) ou sous la barre de progression (Mac OS). Ce paramètre est optionnel.

Passez **Vrai** dans *premierPlan* si vous voulez forcer le passage de la fenêtre de progression au premier plan de l'application.

Exemple

Mise à jour de la progression de la barre :

```
$P:=Progress New // on crée une nouvelle barre
// Effectuer un traitement dans une boucle
Boucle($i;1;100)
// ... code du traitement
// Mise à jour de la barre de progression
$r:=$i/100
Progress SET PROGRESS($P;$r;"More info")
Fin de boucle
// Suppression de la barre une fois le traitement terminé
Progress QUIT($P)
```



Progress SET TITLE

Progress SET TITLE (id ; titre {; progression {; message {; premierPlan}} })

Paramètre	Type		Description
id	Entier long	→	ID de la barre de progression
titre	Texte	→	Titre de la barre
progression	Réel	→	Valeur de progression ([0...1] ou -1)
message	Texte	→	Message de la barre
premierPlan	Booléen	→	Montrer la progression au premier plan

Description

La méthode **Progress SET TITLE** vous permet de définir le titre de la barre de progression ainsi que les informations affichées dans la fenêtre de progression.

Passez dans *id* le numéro d'ID unique de la barre de progression, retourné par la méthode **Progress New**.

Passez dans *titre* le texte principal à afficher dans la fenêtre de progression.

Passez dans *progression* la valeur courante de la barre de progression (facultatif). Vous pouvez passer une valeur réelle (entre 0 et 1) ou -1 afin de définir une barre de progression indéfinie (aussi appelée "Barber shop" sous Mac OS).

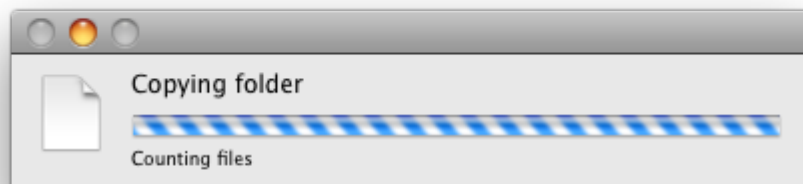
Passez dans *message* un texte additionnel à afficher sous le titre principal (Windows) ou sous la barre de progression (Mac OS). Ce paramètre est optionnel.

Passez **Vrai** dans *premierPlan* si vous voulez forcer le passage de la fenêtre de progression au premier plan de l'application.

Exemple

Création d'une fenêtre de progression simple :

```
$P:=Progress New  
Progress SET TITLE($P;"Copying folder";-1;"Counting files")
```



Progress SET WINDOW VISIBLE

Progress SET WINDOW VISIBLE (visible {; posHor ; posVert {; premierPlan}})

Paramètre	Type		Description
visible	Booléen	→	Vrai = Montrer, Faux = Cacher
posHor	Entier long	→	Coordonnée gauche de la fenêtre -1 = Pas de changement
posVert	Entier long	→	Coordonnée haute de la fenêtre -1 = Pas de changement
premierPlan	Booléen	→	Montrer la progression au premier plan

Description

La méthode **Progress SET WINDOW VISIBLE** vous permet de gérer l'affichage de la fenêtre de progression, si elle existe.

Le paramètre *visible* indique si la fenêtre doit être visible ou non (par défaut, la fenêtre est visible). Passez **Faux** dans ce paramètre pour masquer la fenêtre. Passez **Vrai** pour l'afficher de nouveau si elle existe.

Les paramètres *posHor* et *posVert* permettent de modifier l'emplacement de la fenêtre de progression à l'écran. Vous pouvez passer dans ces paramètres des valeurs représentant le décalage en pixels de la fenêtre vers la droite (*posHor*) ou vers le bas (*posVert*) par rapport aux coordonnées de la fenêtre principale de l'application (sous Windows) ou de l'écran (sous Mac OS).

Pour ne pas modifier les coordonnées (si vous souhaitez utiliser le paramètre *premierPlan* sans déplacer la fenêtre), passez -1 dans chacun de ces paramètres.

Passez **Vrai** dans *premierPlan* si vous voulez forcer le passage de la fenêtre de progression au premier plan de l'application.

Exemple 1

Placer la fenêtre de progression de 50 pixels du bord gauche et 100 pixels du bord haut :

```
Progress SET WINDOW VISIBLE(Vrai;50;100)
```

Exemple 2

Masquer la fenêtre de progression :

```
Progress SET WINDOW VISIBLE(Faux)
```

Exemple 3

Afficher la fenêtre de progression et la faire passer au premier plan sans modifier sa position :

```
Progress SET WINDOW VISIBLE(Vrai;-1;-1;Vrai)
```

Progress Stopped

Progress Stopped (id) -> Résultat

Paramètre	Type	Description
id	Entier long	→ ID de la barre de progression
Résultat	Booléen	↺ Vrai = L'utilisateur a cliqué sur le bouton Stop

Description

La méthode **Progress Stopped** retourne **Vrai** si l'utilisateur a cliqué sur le bouton **Stop** de la barre de progression désignée par le paramètre *id*.

Vous devez appeler cette méthode pour tester si l'utilisateur a cliqué sur le bouton **Stop**. Le bouton ne déclenche pas d'événement lui-même.

Exemple

Exemple de barre de progression sur une boucle :

```
$ProgressID:=Progress New // création d'une nouvelle barre
// La barre doit avoir un bouton Stop
Progress SET BUTTON ENABLED($ProgressID;Vrai)
Boucle($i;1;100)
// Tant que la progression n'est pas stoppée...
Si(Non(Progress Stopped($ProgressID)))
    Progress SET TITLE($ProgressID;"Test progression #"+Chaine($ProgressID))
    Progress SET PROGRESS($ProgressID;$i/100)
    Progress SET MESSAGE($ProgressID;Chaine(100*$i/100)+" %")
    (...)
Sinon // L'utilisateur a cliqué sur Stop
    $i:=100 // On sort de la boucle
Fin de si
Fin de boucle
// Fermeture finale de la barre (le bouton Stop lui-même ne fait rien)
Progress QUIT($ProgressID)
```