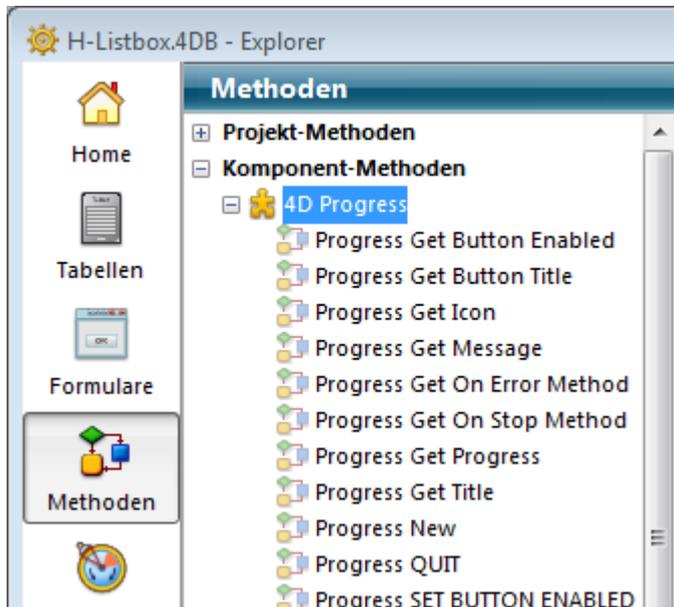


4D Progress

4D bietet eine Komponente mit Namen 4D Progress. Damit können Sie im gleichen Fenster eine oder mehrere Ablaufbalken gleichzeitig öffnen (so wie im Finder auf Mac OS).

Jedem Ablaufbalken ist eine ID zugewiesen, die automatisch über die Methode **Progress New** erzeugt wird. Diese ID verwenden alle Projektmethoden der Komponente, um die Eigenschaften und Werte in Dialogfenstern mit Ablaufbalken zu verwalten.

Die Projektmethoden für diese Komponente erscheinen im Explorer auf der Seite Methoden unter **Komponent-Methoden**:



 Ablaufbalken

 Alphabetische Liste der Befehle

Ablaufbalken

-  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) -> Funktionsergebnis

Parameter	Typ	Beschreibung
id	Lange Ganzzahl	→ ID des Ablaufbalkens
Funktionsergebnis	Boolean	↻ Wahr=Stop Schaltfläche angezeigt, Falsch=Stop Schaltfläche nicht angezeigt

Beschreibung

Die Methode **Progress Get Button Enabled** gibt Wahr zurück, wenn der Ablaufbalken, angegeben im Parameter *id*, die Stop Schaltfläche anzeigt. Zeigt er keine an (StandardEinstellung), gibt sie Falsch zurück.

Progress Get Button Title

Progress Get Button Title (id) -> Funktionsergebnis

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Funktionsergebnis	Text	↩	Titel der Stop Schaltfläche

Beschreibung

Hinweis: Diese Methode ist nur unter Windows verwendbar. Auf Mac OS haben Stop Schaltflächen keine Titel.

Die Methode **Progress Get Button Title** gibt den aktuellen Titel der Stop Schaltfläche für den Ablaufbalken, angegeben im Parameter *id*, zurück.

Der Titel lautet standardmäßig "Stop". Beachten Sie, dass die Methode den aktuellen Titel auch zurückgibt, wenn die Stop Schaltfläche nicht angezeigt wird.

Progress Get Icon

Progress Get Icon (id) -> Funktionsergebnis

Parameter	Typ		Beschreibung
id	Lange Ganzzahl		ID des Ablaufbalkens
Funktionsergebnis	Bild		Icon des Ablaufbalkens

Beschreibung

Die Methode **Progress Get Icon** gibt das Icon des Ablaufbalkens, angegeben im Parameter *id*, zurück.

Progress Get Message

Progress Get Message (id) -> Funktionsergebnis

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Funktionsergebnis	Text	↩	Meldung des Ablaufbalkens

Beschreibung

Die Methode **Progress Get Message** gibt die Meldung des Ablaufbalkens, angegeben im Parameter *id*, zurück.

Progress Get On Error Method

Progress Get On Error Method -> Funktionsergebnis

Parameter	Typ	Beschreibung
Funktionsergebnis	Text	 Name der Methode, die bei einem Fehler aufgerufen wird (wenn angegeben)

Beschreibung

Die Methode **Progress Get On Error Method** gibt den Namen der Projektmethode der Host Datenbank zurück, die aufgerufen wird, wenn beim Verwenden des Ablaufbalkens ein Fehler auftritt.

Ist keine Fehlermethode angegeben, gibt diese Methode einen leeren String zurück.

Progress Get On Stop Method

Progress Get On Stop Method (id) -> Funktionsergebnis

Parameter	Typ	Beschreibung
id	Lange Ganzzahl	→ ID des Ablaufbalkens
Funktionsergebnis	Text	↪ Name der Methode, die beim Anklicken der Stop Schaltfläche aufgerufen wird (wenn angegeben)

Beschreibung

Die Methode **Progress Get On Stop Method** gibt den Namen der Projektmethode der Host Datenbank zurück, die aufgerufen wird, wenn ein Benutzer im Ablaufbalken, angegeben im Parameter *id*, auf die Stop Schaltfläche klickt.

Ist der Stop Schaltfläche keine Methode zugewiesen, gibt diese Methode einen leeren String zurück.

Progress Get Progress

Progress Get Progress (id) -> Funktionsergebnis

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Funktionsergebnis	Zahl	↩	Wert des Ablaufbalkens

Beschreibung

Die Methode **Progress Get Progress** gibt den aktuellen Wert des Ablaufbalkens, angegeben im Parameter *id*, zurück.

Progress Get Title

Progress Get Title (id) -> Funktionsergebnis

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Funktionsergebnis	Text	↩	Titel des Ablaufbalkens

Beschreibung

Die Methode **Progress Get Title** gibt den Haupttitel des Ablaufbalkens, angegeben im Parameter *id*, zurück.

⚙ Progress New

Progress New -> Funktionsergebnis

Parameter	Typ	Beschreibung
Funktionsergebnis	Lange Ganzzahl	➡ ID des neuen Ablaufbalkens

Beschreibung

Die Methode **Progress New** erstellt einen neuen Ablaufbalken und gibt dessen ID Nummer zurück. Diese Nummer ist während der Dauer des Ablaufbalkens einmalig, kann aber anschließend wieder verwendet werden.

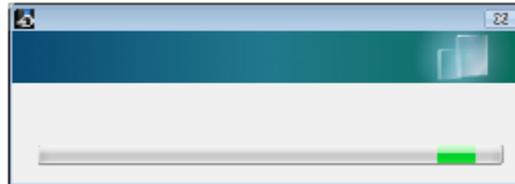
Beim ersten Aufrufen dieser Methode wird ein lokaler Prozess erstellt und über dem Hauptfenster öffnet sich ein neues zentriertes Fenster. Dieses Fenster enthält standardmäßig:

- einen undefinierten Ablaufbalken
- ohne eine Meldung

Mac OS



Windows



⚙ Progress QUIT

Progress QUIT (id)

Parameter	Typ	Beschreibung
id	Lange Ganzzahl	→ ID des Ablaufbalkens

Beschreibung

Die Methode **Progress QUIT** schließt den Ablaufbalken, angegeben im Parameter *id*.

Bezeichnet *id* nur den angezeigten Ablaufbalken, wird das Ablauffenster ebenfalls geschlossen (sowie der lokale Prozess). Andernfalls wird das Fenster angepasst, so dass es nur die noch offenen Ablaufbalken enthält.

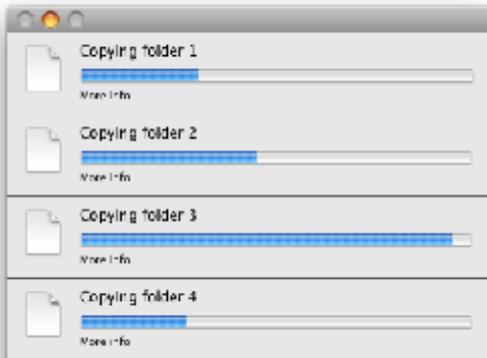
Sie können 0 im Parameter *id* übergeben, um alle Ablaufbalken zu stoppen und das Ablauffenster zu schließen.

Beispiel

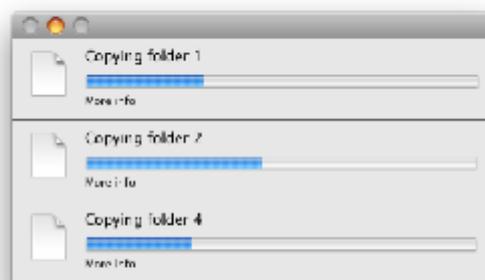
Hat der Balken mit Namen "Copying folder 3" die ID Nummer 3, schreiben Sie:

Progress QUIT(3)

Before

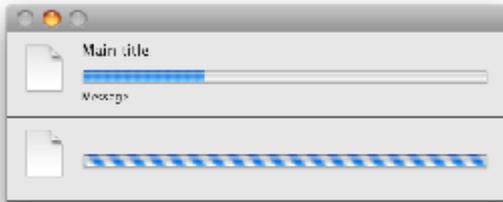


After

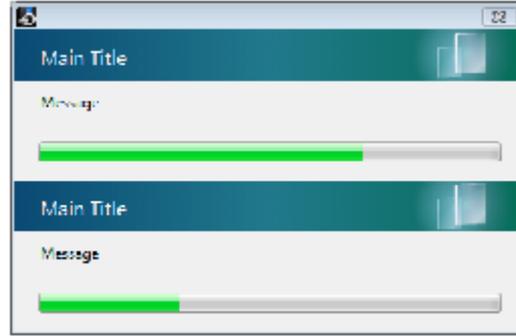


Ist das Ablauffenster beim Öffnen dieser Methode bereits geöffnet, wird es so angepasst, dass es im gleichen Prozess unter dem bzw. den vorigen Balken einen neuen Ablaufbalken anzeigt:

Mac OS



Windows



⚙️ Progress SET BUTTON ENABLED

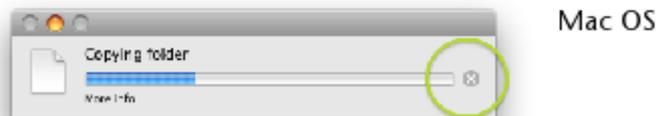
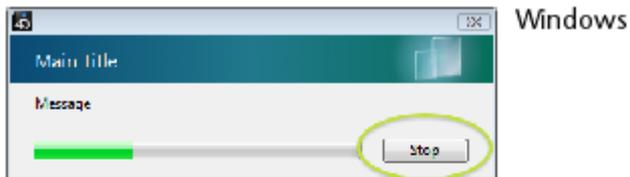
Progress SET BUTTON ENABLED (id ; button)

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	➔	ID des Ablaufbalkens
button	Boolean	➔	Wahr=Anzeigen, Falsch=Ausblenden

Beschreibung

Die Methode **Progress SET BUTTON ENABLED** fügt eine Stop Schaltfläche im Ablaufbalken, angegeben im Parameter *id*, hinzu.

Standardmäßig haben Ablaufbalken keine Stop Schaltflächen. Übergeben Sie im Parameter *Button* wahr, wird eine Schaltfläche angezeigt:



Sie müssen die Auswirkung durch Klicken auf diese Stop Schaltfläche über die Methode **Progress SET ON STOP METHOD** verwalten oder durch Testen des Werts der Methode **Progress Stopped**.

⚙ Progress SET BUTTON TITLE

Progress SET BUTTON TITLE (id ; title)

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
title	Text	→	Titel der Schaltfläche Stop

Beschreibung

Hinweis: Diese Methode ist nur unter Windows verwendbar. Auf Mac OS haben Stop Schaltflächen keine Titel.

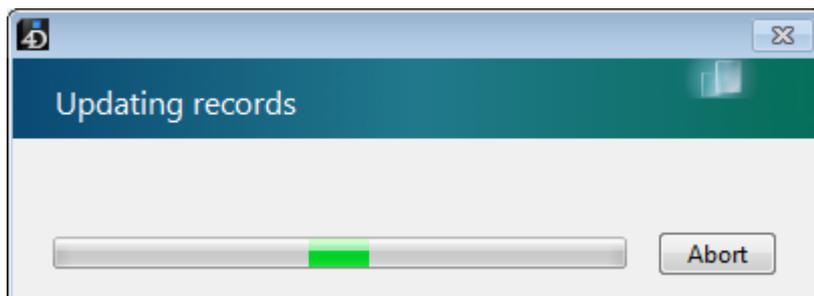
Die Methode **Progress SET BUTTON TITLE** gibt den aktuellen Titel der Stop Schaltfläche für den Ablaufbalken, angegeben im Parameter *id*, zurück.

Der Titel lautet standardmäßig "Stop". Beachten Sie, dass die Methode den aktuellen Titel auch zurückgibt, wenn die Stop Schaltfläche nicht angezeigt wird.

Beispiel

Sie wollen den Titel "Abbrechen" verwenden:

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



⚙ Progress SET FONT SIZES

Progress SET FONT SIZES (sizeTitel {; sizeMeldungen {; sizeButtons}})

Parameter	Typ	Beschreibung
sizeTitel	Lange Ganzzahl	→ Schriftgröße für Titel
sizeMeldungen	Lange Ganzzahl	→ Schriftgröße für Meldungen
sizeButtons	Lange Ganzzahl	→ (Windows)Schriftgröße für Stop Schaltflächen

Beschreibung

Die Methode **Progress SET FONT SIZES** ändert die Schriftgröße für die verschiedenen Texte, die in allen Ablauffenstern erscheinen.

In den Parametern *sizeTitel*, *sizeMeldungen* und *sizeButtons* übergeben Sie die entsprechenden Schriftgrößen. Wollen Sie eine Größe nicht verändern, übergeben Sie im jeweiligen Parameter -1.

Beispiel 1

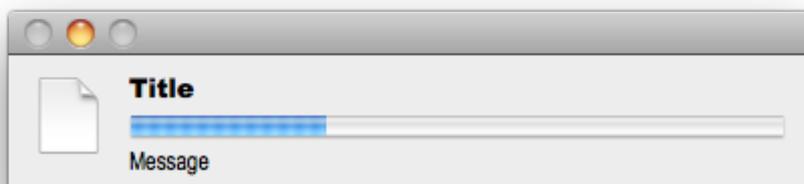
Wir wollen nur die Größe von Meldungen ändern:

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

Beispiel 2

Wir wollen die Größe von Titeln und Meldungen ändern:

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



Progress SET FONTS

Progress SET FONTS (Schrifttitel {; Schriftmeldungen {; Schriftbuttons}})

Parameter	Typ		Beschreibung
Schrifttitel	Text	→	Schrift für Titel
Schriftmeldungen	Text	→	Schrift für Meldungen
Schriftbuttons	Text	→	(Windows) Schrift für Stop Schaltflächen

Beschreibung

Die Methode **Progress SET FONTS** ändert die Schriften für die verschiedenen Texte in allen Ablauffenstern.

In den Parametern *Schrifttitel*, *Schriftmeldungen* und *SchriftButtons* übergeben Sie die Namen der gewünschten Schriften. Wollen Sie eine Schrift nicht ändern, übergeben Sie im jeweiligen Parameter einen leeren String ("").

Beispiel

Wir wollen nur die Schrift für Meldungen ändern:

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

⚙️ Progress SET ICON

Progress SET ICON (id ; icon { ; Vordergrund})

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	➔	ID des Ablaufbalkens
icon	Bild	➔	Bild, welches als Icon erscheinen soll
Vordergrund	Boolean	➔	Ablaufbalken im Vordergrund zeigen

Beschreibung

Die Methode **Progress SET ICON** ändert das im Ablaufbalken angezeigte Icon. Standardmäßig erscheinen folgende Icons:



Windows



Mac OS

In *id* übergeben Sie die einmalige ID des Ablaufbalkens, der von der Methode **Progress New** zurückgegeben wird.

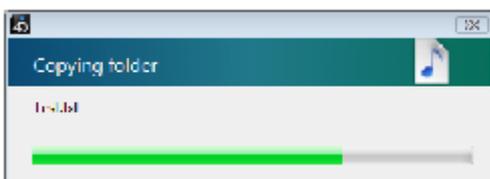
In *icon* übergeben Sie das Bild (Variable oder Feld), das im Fenster mit dem Ablaufbalken als Icon verwendet werden soll. Als maximale Größe gilt:

- auf Mac OS, 40 x 40 Pixel
- unter Windows, 40 x 80 Pixel

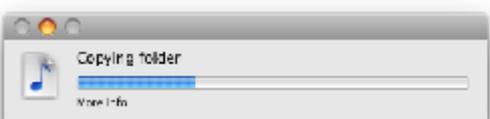
Übergeben Sie ein Icon, das kleiner als diese Begrenzung ist, wird es zentriert aber nicht angepasst. Ist es dagegen größer als diese Begrenzung, wird es zentriert und angepasst.

Übergeben Sie *Wahr* in *Vordergrund*, wenn das Ablauffenster im Vordergrund der Anwendung erscheinen soll.

Beispiele für eigene Icons:



Windows



Mac OS

Progress SET MESSAGE

Progress SET MESSAGE (id ; Meldung {; Vordergrund})

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Meldung	Text	→	Meldung des Ablaufbalkens
Vordergrund	Boolean	→	Ablaufbalken im Vordergrund zeigen

Beschreibung

Die Methode **Progress SET MESSAGE** ändert die Meldung im Ablaufbalken.

In *id* übergeben Sie die einmalige ID des Ablaufbalkens, welche die Methode **Progress New** zurückgibt.

In *Meldung* übergeben Sie den Text, den Sie unter dem Haupttitel (Windows) oder unter dem Ablauffenster (Mac OS) ändern wollen.

Übergeben Sie *Wahr* in *Vordergrund*, wenn das Ablauffenster zwingend im Vordergrund der Anwendung erscheinen soll.

⚙️ Progress SET ON ERROR METHOD

Progress SET ON ERROR METHOD (*MethodenName*)

Parameter	Typ		Beschreibung
MethodenName	Text	➔	Name der Fehlermethode

Beschreibung

Die Methode **Progress SET ON ERROR METHOD** bestimmt eine Methode, die beim Verwenden des Ablaufbalkens im Falle eines Fehlers ausgeführt werden soll, z.B. id unbekannt, inkorrekte Anzahl Parameter, etc.).

In *MethodenName* übergeben Sie den Namen der Projektmethode der Host Datenbank, die im Falle eines Fehlers aufgerufen werden soll. Diese Methode wird von allen Ablau fenstern der Anwendung gemeinsam genutzt.

Hinweis: Da die Methode **Progress SET ON ERROR METHOD** zu einer Komponente gehört, müssen Sie der Methode *MethodenName* die Eigenschaft "Gemeinsam von Komponenten und Host benutzt" zuweisen; andernfalls wird ein Fehler zurückgegeben.

Die Methode *MethodenName* wird mit drei Parametern aufgerufen:

- \$1 (Lange Ganzzahl): Fehlernummer
- \$2 (Text): Fehlertext
- \$3 (Lange Ganzzahl): Einmalige ID des Ablaufbalkens

Beispiel

Beispiel für eine Methode, die im Falle eines Fehlers aufgerufen wird. Diese Methode wird im Designmodus als "gemeinsam genutzt" deklariert:

```
C_LONGINT($1)
```

```
C_TEXT($2)
```

```
C_LONGINT($3)
```

```
C_LONGINT($ErrorID)
```

```
C_TEXT($ErrorText)
```

```
C_LONGINT($ProgressID)
```

```
$ErrorID:=$1
```

```
$ErrorText:=$2
```

```
$ProgressID:=$3
```

```
$Error:=""
```

```
$Error:=$Error+"Error number: "+String($ErrorID)+Char(Carriage return)
```

```
$Error:=$Error+$ErrorText+Char(Carriage return)
```

```
$Error:=$Error+"Progress ID: "+String($ProgressID)  
ALERT($Error)
```

Progress SET ON STOP METHOD

Progress SET ON STOP METHOD (id ; MethodenName)

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID of progress bar
MethodenName	Text	→	Name der Stop Methode

Beschreibung

Die Methode **Progress SET ON STOP METHOD** bestimmt eine Methode, die ausgeführt werden soll, wenn der Benutzer im Ablaufbalken auf die Stop Schaltfläche klickt.

Standardmäßig haben Ablaufbalken keine Stop Schaltflächen. Soll ein Ablaufbalken diese Schaltfläche haben, müssen Sie die Methode **Progress SET BUTTON ENABLED** verwenden.

In *id* übergeben Sie die einmalige ID des Ablaufbalkens, welche die Methode **Progress New** zurückgibt.

In *MethodenName* übergeben Sie den Namen der Projektmethode der Host Datenbank, die beim Anklicken der Stop Schaltfläche aufgerufen werden soll. Diese Methode empfängt die einmalige ID des Ablaufbalkens als ersten Parameter. Sie wird dann in einem von der Komponente gestarteten neuen Prozess ausgeführt.

Hinweis: Da die Methode **Progress SET ON STOP METHOD** zu einer Komponente gehört, müssen Sie der Methode *MethodenName* die Eigenschaft "Gemeinsam von Komponenten und Host benutzt" zuweisen; andernfalls wird ein Fehler zurückgegeben.

Progress SET PROGRESS

Progress SET PROGRESS (id ; Ablauf {; Meldung {; Vordergrund} })

Parameter	Typ	Beschreibung
id	Lange Ganzzahl	→ ID des Ablaufbalkens
Ablauf	Zahl	→ Wert des Ablaufs ([0...1] oder -1)
Meldung	Text	→ Meldung des Ablaufbalkens
Vordergrund	Boolean	→ Ablaufbalken im Vordergrund zeigen

Beschreibung

Die Methode Progress SET PROGRESS ändert den Wert des Ablaufbalkens zusammen mit der Information, die im Ablauffenster erscheint. Es ist hilfreich, einen Ablaufbalken innerhalb einer Schleife zu aktualisieren.

In *id* übergeben Sie die einmalige ID des Ablaufbalkens, welche die Methode **Progress New** zurückgibt.

In *Ablauf* übergeben Sie den aktuellen Wert des Ablaufbalkens. Sie übergeben einen Wert Zahl (zwischen 0 und 1) oder -1 für einen undefinierten Ablaufbalken (auf Mac OS bekannt als "animierter Balken").

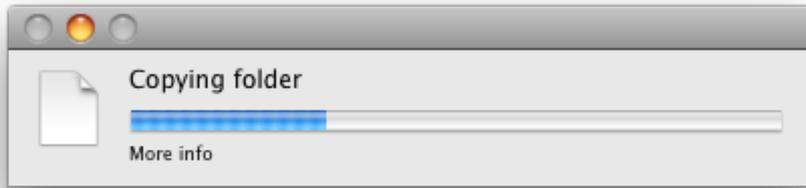
In *Meldung* übergeben Sie zusätzlichen Text, der unter dem Haupttitel (Windows) oder unter dem Ablaufbalken (Mac OS) erscheinen soll. Dieser Parameter ist optional.

Übergeben Sie *Wahr* in *Vordergrund*, wenn das Ablauffenster zwingend im Vordergrund der Anwendung erscheinen soll.

Beispiel

Ablaufbalken updaten:

```
$P:=Progress New // Neuen Balken erstellen
// Ablufen in einer Schleife ausführen
For($i;1;100)
// ... Code für Ablufen
// Ablaufbalken updaten
    $r:=$i/100
    Progress SET PROGRESS($P;$r;"Mehr Info")
End for
// Balken löschen, wenn Ablufen beendet ist
PROGRESS QUIT($P)
```



⚙ Progress SET TITLE

Progress SET TITLE (id ; Titel {; Ablauf {; Meldung {; Vordergrund}} })

Parameter	Typ		Beschreibung
id	Lange Ganzzahl	→	ID des Ablaufbalkens
Titel	Text	→	Titel des Ablaufbalkens
Ablauf	Zahl	→	Wert des Ablaufs ([0...1] oder -1)
Meldung	Text	→	Meldung des Ablaufbalkens
Vordergrund	Boolean	→	Zeige Ablaufbalken im Vordergrund

Beschreibung

Die Methode **Progress SET TITLE** setzt den Titel des Ablaufbalkens zusammen mit der Information, die im Ablauffenster erscheint.

In *id* übergeben Sie die einmalige ID des Ablaufbalkens, welche die Methode **Progress New** zurückgibt.

In *Titel* übergeben Sie den Haupttext, der im Fenster mit Ablaufbalken erscheinen soll.

In *Ablauf* übergeben Sie den aktuellen Wert des Ablaufbalkens (optional). Sie übergeben einen Wert Zahl (zwischen 0 und 1) oder -1 für einen undefinierten Ablaufbalken (auf Mac OS auch bekannt als "animierter Balken").

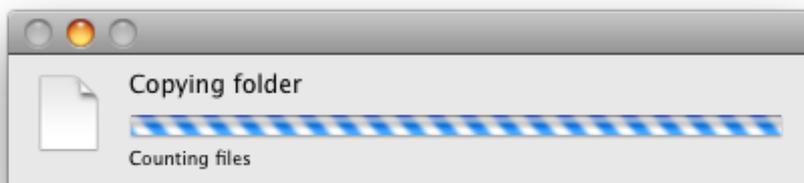
In *Meldung* übergeben Sie zusätzlichen Text, der unter dem Haupttitel (Windows) oder unter dem Ablaufbalken (Mac OS) erscheinen soll. Dieser Parameter ist optional.

Übergeben Sie *Wahr* in *Vordergrund*, wenn das Ablauffenster zwingend im Vordergrund der Anwendung erscheinen soll.

Beispiel

Einfaches Fenster mit Ablaufbalken erstellen:

```
$P:=Progress New  
Progress SET TITLE($P;"Ordner kopieren";-1;"Dateien zählen")
```



Progress SET WINDOW VISIBLE

Progress SET WINDOW VISIBLE (Sichtbar {; horPos ; vertPos {; Vordergrund}})

Parameter	Typ	Beschreibung
Sichtbar	Boolean	→ Wahr=Anzeigen, Falsch=Ausblenden
horPos	Lange Ganzzahl	→ Linke Koordinate des Fensters, -1 = Keine Änderung
vertPos	Lange Ganzzahl	→ Obere Koordinate des Fensters, -1 = Keine Änderung
Vordergrund	Boolean	→ Zeige Ablaufbalken im Vordergrund

Beschreibung

Die Methode **Progress SET WINDOW VISIBLE** verwaltet die Anzeige des Fensters mit dem Ablaufbalken, wenn es existiert.

Der Parameter *Sichtbar* gibt an, ob das Fenster sichtbar ist. Standardmäßig ist es sichtbar. Übergeben Sie *Falsch*, um das Fenster auszublenden, *Wahr*, um es wieder einzublenden, wenn es existiert.

Die Parameter *horPos* und *vertPos* ändern die Position des Fensters mit Ablaufbalken auf dem Bildschirm. In diesen Parametern übergeben Sie Werte für das Verschieben in Pixel nach rechts (*horPos*) oder nach unten (*vertPos*) in Bezug auf die Koordinaten des Hauptfensters der Anwendung (unter Windows) oder des Bildschirms (auf Mac OS).

Wollen Sie diese Koordinate nicht verändern, z.B. wenn Sie den Parameter *Vordergrund* ohne Verändern der Position des Fensters verwenden wollen, übergeben Sie in beiden Parametern -1.

Übergeben Sie *Wahr* in *Vordergrund*, wenn das Ablauffenster zwingend im Vordergrund der Anwendung erscheinen soll.

Beispiel 1

Fenster mit Ablaufbalken 50 Pixel vom linken Rand und 100 Pixel vom oberen Rand setzen:

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

Beispiel 2

Fenster mit Ablaufbalken ausblenden:

```
Progress SET WINDOW VISIBLE(False)
```

Beispiel 3

Fenster mit Ablaufbalken anzeigen und in den Vordergrund setzen, ohne seine Position zu verändern:

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

⚙️ Progress Stopped

Progress Stopped (id) -> Funktionsergebnis

Parameter	Typ	Beschreibung
id	Lange Ganzzahl	➔ ID des Ablaufbalkens
Funktionsergebnis	Boolean	➔ Wahr = Benutzer hat auf Stop Schaltfläche geklickt

Beschreibung

Die Methode **Progress Stopped** gibt *Wahr* zurück, wenn ein Benutzer auf die Stop Schaltfläche des Ablaufbalkens, angegeben im Parameter *id*, klickt. Sie müssen diese Methode aufrufen, um zu testen, ob der Benutzer auf die Stop Schaltfläche geklickt hat. Die Schaltfläche löst nicht das Ereignis ansich aus.

Beispiel

Beispiel für einen Ablaufbalken in einer Schleife:

```
$ProgressID:=Progress New // Neuen Ablaufbalken erstellen
// Der Ablaufbalken muss eine Stop Schaltfläche haben
Progress SET BUTTON ENABLED($ProgressID;True)
For($i;1;100)
// o lange der Ablauf nicht gestoppt wird...
  If(Not(Progress Stopped($ProgressID)))
    Progress SET TITLE($ProgressID;"Test progress #"+String($ProgressID))
    Progress SET PROGRESS($ProgressID;$i/100)
    Progress SET MESSAGE($ProgressID;String(100*$i/100)+" %")
    (...)
  Else // Der Benutzer klickt auf Stop
    $i:=100 // Wir verlassen die Schleife
  End if
End for
// Ablaufbalken wird endgültig geschlossen (die Stop Schaltfläche selbst tut nichts)
Progress QUIT($ProgressID)
```