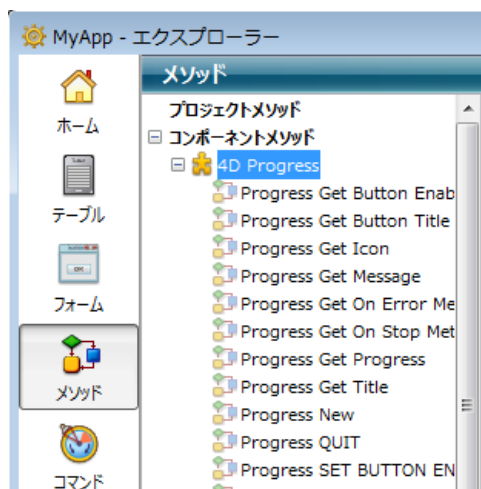


4D Progress (進捗バー)

4Dには新しい組み込みコンポーネント、4D Progressが含まれます。このコンポーネントを使用して (Mac OSのFinderのように) 1つ以上の進捗バーを同じウィンドウに表示することができます。

各進捗バーには**Progress New**メソッドから自動でIDが割り当てられます。このIDは進捗ダイアログボックスの属性と値を管理するコンポーネントメソッドで使用されます。























このコンポーネントの共有プロジェクトメソッドはエクスプローラーの**コンポーネントメソッド**ページで見ることができます:



📄 4D Progressメソッドリスト

📄 コマンドリスト (文字順)

4D Progressメソッドリスト

-  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) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	ブール	↩	True = 停止ボタンを表示; False = 停止ボタンを非表示

説明

Progress Get Button Enabledメソッドは`id`引数で指定された進捗バーに停止ボタンが表示されている場合、**True**を返します。表示されない場合 (デフォルトオプション) **False**を返します。

Progress Get Button Title

Progress Get Button Title (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	テキスト	↩	停止ボタンのラベル

説明

注: このメソッドはWindowsでのみ使用できます。では停止ボタンにラベルが表示されません。

Progress Get Button Titleメソッドは`id`引数で指定された進捗バーの停止ボタンのラベルを返します。

デフォルトのラベルは"停止"です。メソッドは停止ボタンが表示されていない場合でも設定されているラベルを返す点に留意してください。

Progress Get Icon

Progress Get Icon (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	ピクチャー	↩	進捗バーのアイコン

説明

Progress Get Icon メソッドは *id* 引数で指定された進捗バーのアイコンを返します。

Progress Get Message

Progress Get Message (id) -> 戻り値


引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	テキスト	↩	進捗バーメッセージ

説明

Progress Get Messageメソッドは`id`引数で指定された進捗バーのメッセージを返します。

Progress Get On Error Method

Progress Get On Error Method -> 戻り値

引数	型	説明
戻り値	テキスト	 エラーが発生したときに呼び出されるメソッド (指定した場合)

説明

Progress Get On Error Methodは進捗バー使用時エラーが発生した場合に呼び出される、ホストデータベースのプロジェクトメソッド名を返します。

エラーメソッドが指定されていない場合、メソッドは空の文字列を返します。

Progress Get On Stop Method

Progress Get On Stop Method (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	テキスト	↩	停止ボタンがクリックされたときに呼び出されるメソッド (指定した場合)

説明

Progress Get On Stop Methodは`id`引数で指定された進捗バーの停止ボタンがクリックされたときに呼び出される、ホストデータベースのプロジェクトメソッド名を返します。

停止ボタンにメソッドが割り当てられていない場合、メソッドは空の文字列を返します。

Progress Get Progress

Progress Get Progress (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	実数	↩	進捗バーの値

説明

Progress Get Progressメソッドは`id`引数で指定した進捗バーに関連付けられている現在の値を返します。

Progress Get Title

Progress Get Title (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	テキスト	↩	進捗バーのタイトル

説明

Progress Get Titleメソッドは`id`引数で指定された進捗バーのメインタイトルを返します。

Progress New -> 戻り値

引数	型		説明
戻り値	倍長整数		進捗バーのID

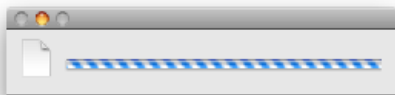
説明

Progress Newメソッドは新しい進捗バーを作成し、そのID番号を返します。このIDは進捗バーが表示されている間はユニークですが、閉じられた後は再利用されることがあります。

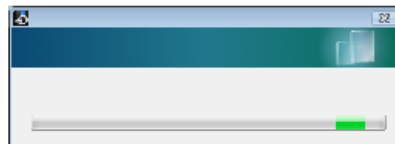
最初にこのメソッドが呼び出されるとローカルプロセスが作成され、メインウィンドウ中央に新しいウィンドウが開かれます。デフォルトでこのウィンドウは:

- 未定義の進捗バーが表示されます。
- メッセージも表示されません。

Mac OS

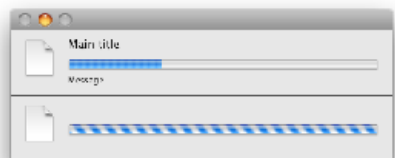


Windows

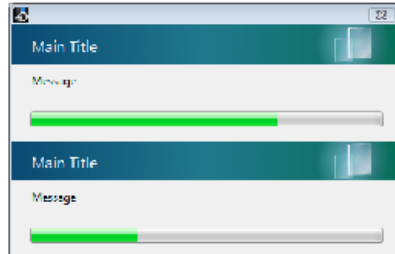


このメソッドが呼び出されたとき、すでに進捗ウィンドウが開かれていれば、同じプロセス内で新しい進捗バーが一番下に表示されるよう、ウィンドウがリサイズされます:

Mac OS



Windows



Progress QUIT

Progress QUIT (id)

引数	型	説明
id	倍長整数	進捗バーのID

説明

Progress QUITメソッドは`id`引数で指定された進捗バーを閉じます。

`id`で指定された進捗バーが表示されている唯一のものであれば、ウィンドウ、およびローカルプロセスも閉じられます。そうでない場合表示中の進捗バーだけが表示されるよう、ウィンドウがリサイズされます。

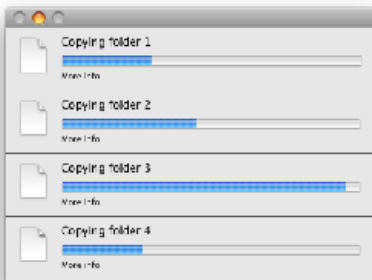
`id` 引数に0を渡す事で、全ての進捗バーと進捗ウィンドウを閉じる事ができます。

例題

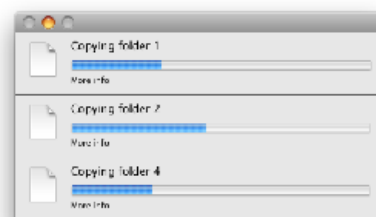
"フォルダー3をコピー中"進捗バーのIDが3のとき:

Progress QUIT(3)

Before



After



⚙️ Progress SET BUTTON ENABLED

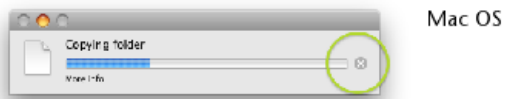
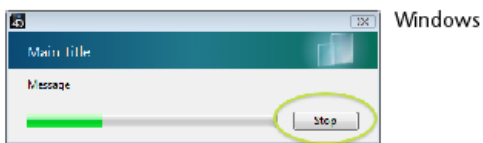
Progress SET BUTTON ENABLED (id ; button)

引数	型		説明
id	倍長整数	➡	進捗バーのID
button	ブール	➡	True = 表示, False = 非表示

説明

Progress SET BUTTON ENABLEDメソッドは`id`で指定された進捗バーの**停止ボタン**の表示/非表示を設定するために使用します。

デフォルトで進捗バーには停止ボタンが表示されません。`button`引数に**True**を渡すとこのボタンが表示されます:



停止ボタンが押されたときの動作は**Progress SET ON STOP METHOD**メソッドを使用して、または**Progress Stopped**メソッドの戻り値をテストして、開発者が管理しなくてはなりません。

Progress SET BUTTON TITLE

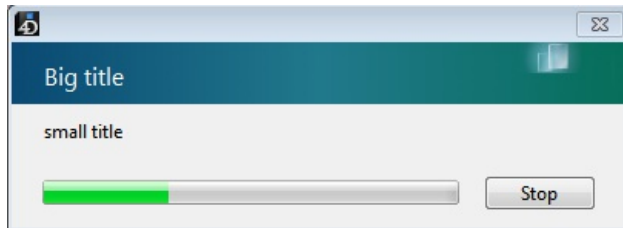
Progress SET BUTTON TITLE (id ; title)

引数	型		説明
id	倍長整数	→	進捗バーのID
title	テキスト	→	停止ボタンのラベル (Windows)

説明

注: このメソッドはWindowsのみで利用できます。Mac OSでは停止ボタンにラベルがありません。

Progress SET BUTTON TITLEメソッドはid引数で指定された進捗バーの**停止ボタン**のラベルを設定するために使用します。デフォルトでこのボタンのラベルは"停止"です:

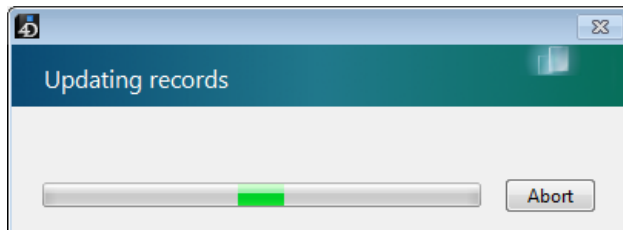


進捗バーの**停止ボタン**はデフォルトでは表示されません。進捗バーにこのボタンを表示させたい場合、**Progress SET BUTTON ENABLED**メソッドを使用しなければなりません。

例題

ボタンラベルに"アボート"を使用したい場合:

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



Progress SET FONT SIZES

Progress SET FONT SIZES (sizeTitles {; sizeMessages {; sizeButtons} })

引数	型		説明
sizeTitles	倍長整数	→	タイトル用のフォントサイズ
sizeMessages	倍長整数	→	メッセージ用のフォントサイズ
sizeButtons	倍長整数	→	(Windows) 停止ボタンのフォントサイズ

説明

Progress SET FONT SIZESメソッドはすべての進捗ウィンドウに適用される、各種テキストのフォントサイズを変更するために使用します。

sizeTitles、*sizeMessages*、そして*sizeButtons*引数には使用するフォントサイズを渡します。サイズを変更したくない場合、対応する引数に-1を渡します。

例題 1

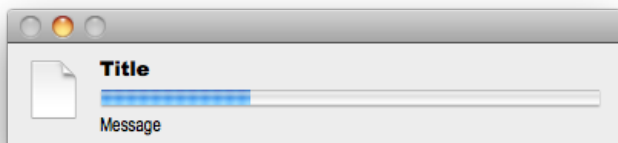
メッセージのサイズのみを変更する場合:

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

例題 2

タイトルおよびメッセージのサイズを変更する場合:

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



Progress SET FONTS

Progress SET FONTS (fontTitles {; fontMessages {; fontButtons} })

引数	型		説明
fontTitles	テキスト	→	タイトル用のフォント
fontMessages	テキスト	→	メッセージ用のフォント
fontButtons	テキスト	→	(Windows) 停止ボタン用のフォント

説明

Progress SET FONTS はすべての進捗バーウィンドウで使用される各種フォント名を変更するために使用します。

fontTitles、*fontMessages*、および*fontButtons*引数には使用するフォント名を渡します。フォント名を変更したくない場合、対応する引数に空の文字列 ("") を渡します。

例題

メッセージ用のフォントのみを変更する場合:

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


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

引数	型		説明
id	倍長整数	→	進捗バーのID
icon	ピクチャー	→	アイコンとして表示するピクチャー
foreground	ブール	→	進捗バーを最前面に表示

説明

Progress SET ICONメソッドは進捗バー内に表示するアイコンを変更するために使用します。デフォルトで以下のアイコンが表示されます:



Windows



Mac OS

*id*には**Progress New**メソッドから返される進捗バーユニークIDを渡します。

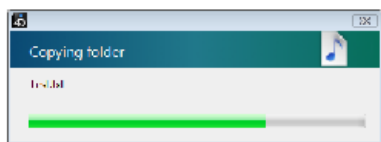
*icon*には進捗バーウィンドウのアイコンとして使用するピクチャーを渡します。このアイコンの最大サイズは:

- Mac OS: 40 x 40 pixels
- Windows: 40 x 80 pixels

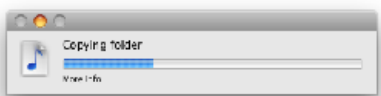
これより小さなアイコンを渡した場合、そのままのサイズで中央に表示されます。制限より大きなアイコンが渡された場合、リサイズされて中央に表示されます。

*foreground*に**True**を渡すと、進捗バーはアプリケーションの最前面に表示されます。

カスタムアイコンの例:



Windows



Mac OS

Progress SET MESSAGE

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

引数	型		説明
id	倍長整数	→	進捗バーのID
message	テキスト	→	進捗バーに表示するメッセージ
foreground	ブール	→	進捗バーを最前面に表示

説明

Progress SET MESSAGEメソッドを使用して進捗バーに表示されるメッセージを変更できます。

*id*には**Progress New**から返される進捗バーユニークIDを渡します。

*message*には (Windowsの場合) メインタイトルの下、あるいは (Mac OSの場合) 進捗バーの下に表示に表示するテキストを渡します。

*foreground*に**True**を渡すと、進捗バーはアプリケーションの最前面に表示されます。

Progress SET ON ERROR METHOD

Progress SET ON ERROR METHOD (methodName)

引数	型	説明
methodName	テキスト	エラーメソッド名

説明

Progress SET ON ERROR METHODメソッドを使用して、進捗バー使用時にエラー (例えば未知のID、引数の数が正しくない等) が発生した場合に実行するメソッドを指定できます。

*methodName*にはエラー発生時に実行する、ホストデータベースのメソッド名を渡します。このメソッドはアプリケーションのすべての進捗ウィンドウ共通です。

注: 進捗バーメソッドはコンポーネントのコンテキストで実行されるため、*methodName*に渡すメソッドは"コンポーネントとホストデータベースで共有する"プロパティが選択されていなければなりません。そうされていない場合、エラーが返されます。

*methodName*メソッドが実行されるときには引数が3つ渡されます:

- \$1 (倍長整数): エラー番号
- \$2 (テキスト): エラーテキスト
- \$3 (倍長整数): 進捗バーユニークID

例題

エラー処理メソッドの例は以下の通りです:

```
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+"エラー番号: "+String($ErrorID)+"\r"
$Error:=$Error+$ErrorText+"\r"
$Error:=$Error+"進捗ID: "+String($ProgressID)
ALERT ($Error)
```

Progress SET ON STOP METHOD

Progress SET ON STOP METHOD (id ; methodName)

引数	型		説明
id	倍長整数	→	進捗バーのID
methodName	テキスト	→	停止ボタンに割り当てるメソッド

説明

Progress SET ON STOP METHODメソッドは、ユーザーが進捗バーの**停止**ボタンをクリックしたときに実行されるメソッドを指定するために使用します。

デフォルトで進捗バーには停止ボタンが表示されません。ボタンを表示させたい場合**Progress SET BUTTON ENABLED**メソッドを使用します。

*id*には**Progress New**メソッドから返される進捗バーIDを渡します。

*methodName*には停止ボタンがクリックされたときに呼び出される、ホストデータベースのプロジェクトメソッド名を渡します。このメソッドが実行されるときには第一引数に進捗バーのユニークIDが渡されます。またこのメソッドはコンポーネントから起動される新規プロセス内で実行されます。

注: 停止メソッドはコンポーネントのコンテキストで実行されるため、*methodName*に渡すメソッドは"コンポーネントとホストデータベースで共有する"プロパティが選択されていなければなりません。そうされていない場合、エラーが返されます。

Progress SET PROGRESS

```
Progress SET PROGRESS ( id ; progress {; message {; foreground} } )
```

引数	型		説明
id	倍長整数	→	進捗バーのID
progress	実数	→	進捗の値 ([0...1] または -1)
message	テキスト	→	進捗バーのメッセージ
foreground	ブール	→	進捗バーを最前面に表示

説明

Progress SET PROGRESS メソッドは進捗バーの値を変更し、進捗ウィンドウに表示されるメッセージを更新するために使用します。特にこのメソッドはループ内で進捗バーを更新するために使用します。

*id*には**Progress New**メソッドから返される進捗バーIDを渡します。

*progress*には進捗バーの現在値を渡します。0から1の間の実数値を渡せます。未定義の進捗バー (Mac OSのバーバーストップタイプ) の場合-1を渡します。

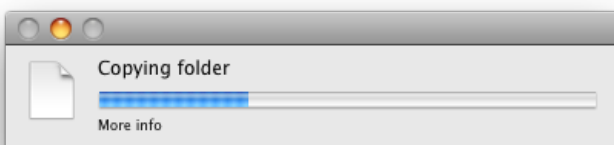
*message*には (Windowsの場合) メインタイトルの下、あるいは (Mac OSの場合) 進捗バーの下に表示に表示するテキストを渡します。この引数はオプションです。

*foreground*に**True**を渡すと、進捗バーはアプリケーションの最前面に表示されます。

例題

進捗バーの更新:

```
$P:=Progress New // 新規バーを作成
// ループ中で処理を実行
For ($i;1;100)
// ... 処理を行うコード
// 進捗バーを更新
    $r:=$i/100
    Progress SET PROGRESS($P;$r;"More info")
End for
// 処理が終了したら進捗バーを閉じる
Progress QUIT($P)
```



Progress SET TITLE

```
Progress SET TITLE ( id ; title {; progress {; message {; foreground}}})
```

引数	型		説明
id	倍長整数	→	進捗バーのID
title	テキスト	→	進捗バーのタイトル
progress	実数	→	進捗バーの値 ([0...1] または -1)
message	テキスト	→	進捗バーのメッセージ
foreground	ブール	→	進捗バーを最前面に表示

説明

Progress SET TITLE メソッドは主に進捗バーのタイトルを設定するために使用します。その他進捗バーウィンドウに表示する情報も設定できます。

*id*には**Progress New**メソッドから返される進捗バーIDを渡します。

*title*には進捗バーウィンドウに表示するメインテキストを渡します。

*progress*には進捗バーの現在値を渡します (オプション)。0 から1の間の実数値を渡せます。未定義の進捗バー (Mac OSのバーバーストップタイプ) の場合-1を渡します。

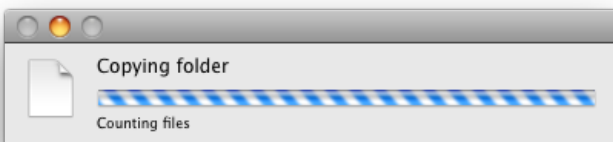
*message*には (Windowsの場合) メインタイトルの下、あるいは (Mac OSの場合) 進捗バーの下に表示に表示するテキストを渡します。この引数はオプションです。

*foreground*に**True**を渡すと、進捗バーはアプリケーションの最前面に表示されます。

例題

シンプルな進捗バーウィンドウを作成:

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



Progress SET WINDOW VISIBLE

Progress SET WINDOW VISIBLE (visible {; horPos ; vertPos {; foreground} })

引数	型		説明
visible	ブール	⇒	True = 表示, False = 非表示
horPos	倍長整数	⇒	ウィンドウの左座標 -1 = 変更しない
vertPos	倍長整数	⇒	ウィンドウの上座標 -1 = 変更しない
foreground	ブール	⇒	進捗バーを最前面に表示

説明

Progress SET WINDOW VISIBLEメソッドを使用して、既存の進捗バーウィンドウの表示プロパティを管理できます。
*visible*引数を使用してウィンドウの表示/非表示を指定できます (デフォルトで表示)。この引数に**False**を渡すとウィンドウが非表示になり、**True**を渡すと表示されます。

*horPos*と*vertPos*引数には画面上での進捗バーウィンドウの位置を指定する値を渡します。この値は (Windowsの場合) メインアプリケーションウィンドウ、または (Mac OSの場合) スクリーンの左上からのピクセル単位の指定です。

座標を変更したくない場合、各引数に-1を渡します。

*foreground*に**True**を渡すと、進捗バーはアプリケーションの最前面に表示されます。

例題 1

進捗バーウィンドウを左50ピクセル、上100ピクセルの位置に表示:

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

例題 2

進捗バーウィンドウを隠す:

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

例題 3

進捗バーウィンドウを表示し、座標は変更せずに最前面にする:

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

Progress Stopped

Progress Stopped (id) -> 戻り値

引数	型		説明
id	倍長整数	→	進捗バーのID
戻り値	ブール	↩	True = ユーザーが停止ボタンをクリックした

説明

Progress Stoppedメソッドは、*id*引数で指定した進捗バーの**停止ボタン**をユーザーがクリックした場合**True**を返します。ユーザーが停止ボタンをクリックしたかどうかをテストする必要がある場合、このメソッドを呼び出します。ボタン自身はなにも行わない点に留意してください。

例題

ループ内での進捗バーの例題:

```
$ProgressID:=Progress New // 新規進捗バーを作成
// 進捗バーに停止ボタンを表示
Progress SET BUTTON ENABLED($ProgressID;True)
For ($i;1;100)
// 停止ボタンがクリックされていなければ繰り返す...
If (Not (Progress Stopped($ProgressID)))
Progress SET TITLE($ProgressID;"Test progress #"+String($ProgressID))
Progress SET PROGRESS($ProgressID;$i/100)
Progress SET MESSAGE($ProgressID;String(100*$i/100)+" %")
(...)
Else // ユーザーが停止ボタンをクリックした
$i:=100 // ループを抜ける
End if
End for
// 進捗バーを閉じる (停止ボタンはなにも行わない)
Progress QUIT($ProgressID)
```

4D Progress (進捗バー) - コマンドリスト (文字順)

P

- 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