

prog_bar

Allan Savage

COLLABORATORS

	<i>TITLE :</i> prog_bar		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Allan Savage	June 4, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	prog_bar	1
1.1	Prog_Bar v1.1 Function List	1
1.2	CreateProgBarA() / CreateProgBar()	1
1.3	SetProgBarAttrsA() / SetProgBarAttrs()	2
1.4	FreeProgBar()	4
1.5	RefreshProgBar()	4
1.6	UpdateProgBar()	5
1.7	ResetProgBar()	5
1.8	ClearProgBar()	6
1.9	ClearBar()	7
1.10	ClearText()	7

Chapter 1

prog_bar

1.1 Prog_Bar v1.1 Function List

```
CreateProgBarA ()
SetProgBarAttrsA ()
FreeProgBar ()
RefreshProgBar ()
UpdateProgBar ()
ResetProgBar ()
ClearProgBar ()
ClearBar ()
ClearText ()
```

1.2 CreateProgBarA() / CreateProgBar()

NAME

```
CreateProgBarA -- Allocates and initialises a progress bar. (V1.0)
CreateProgBar -- Varargs stub for CreateProgBarA(). (V1.0)
```

SYNOPSIS

```
bar = CreateProgBarA ( Wnd, Left, Top, Width, Height, Size,
                      taglist )
```

```
PBAR *CreateProgBarA ( struct Window *, UWORD, UWORD, UWORD, UWORD,
                      UWORD, struct TagList * )
```

```
bar = CreateProgBar ( Wnd, Left, Top, Width, Height, Size,
                     First_Tag, ... )
```

```
PBAR *CreateProgBar ( struct Window *, UWORD, UWORD, UWORD, UWORD,
                     UWORD, Tag, ... )
```

FUNCTION

Allocates a new PBAR structure, initialises it to the default values, or any values specified in the function call, and then renders the progress bar to the specified window.

INPUTS

Wnd - A pointer to the window in which the progress bar is to be rendered.
 Left - The X position of the progress bar.
 Top - The Y position of the progress bar.
 Width - The Width of the progress bar.
 Height - The Height of the progress bar.
 Size - The logical size of the progress bar. This is the value represented by the full bar. It is not connected to the visual dimensions of the progress bar.
 taglist - A pointer to an array of tags specifying optional parameters which will modify the behaviour of the progress bar. If the default values are all that are required this can be NULL. See SetProgBarAttrs() for more information about the various tags available.

RESULT

bar - If successful these functions will return a pointer to the new progress bar. This pointer should be kept as it will be needed to identify the progress bar for all the other functions. These functions will only fail due to a lack of memory, in which case the returned value will be NULL.

BUGS

None known

SEE ALSO

SetProgBarAttrs() FreeProgBar() RefreshProgBar() UpdateProgBar()

1.3 SetProgBarAttrsA() / SetProgBarAttrs()

NAME

SetProgBarAttrsA -- Changes the behaviour or appearance of the progress bar. (V1.0)
 SetProgBarAttrs -- Varargs stub for SetProgBarAttrs(). (V1.0)

SYNOPSIS

```

SetProgBarAttrsA ( PBar, taglist )

void SetProgBarAttrsA ( PBAR *, struct TagList * )

SetProgBarAttrs ( PBar, First_Tag, ... )

void SetProgBarAttrs ( PBAR *, Tag, ... )
  
```

FUNCTION

Changes the behaviour or appearance of a progress bar to that specified by the optional tags. It will perform all the necessary calculations and redraw the display.

INPUTS

PBar - A pointer to the progress bar to be altered. This is the pointer returned by CreateProgBar().
 taglist - A pointer to a tag list. The available tags are as follows.

PB_LeftEdge	The X position of the progress bar.
PB_TopEdge	The Y position of the progress bar.
PB_Width	The Width of the progress bar.
PB_Height	The Height of the progress bar.
PB_Direction	The Direction in which the progress bar will grow. Available options are
PBDE_RIGHT	From Left to Right (default).
PBDE_LEFT	From Right to Left.
PBDE_UP	From Bottom to Top.
PBDE_DOWN	From Top to Bottom.
PB_BarColour	The pen number of the bar's foreground colour. The default value is 3 (Blue).
PB_BarBackColour	The Pen number of the bar's background colour. The default value is 0 (Grey).
PB_BarSize	The logical size of the progress bar.
PB_BarValue	The current Value of the progress bar. It is better to use UpdateProgBar() to change this value if this is the only change. However, the option has been supplied so that the value can be altered at the same time as other changes are made. If the value supplied is greater than PB_BarSize the actual value stored will be PB_BarSize. This guarantees that the bar cannot overflow.
PB_BorderType	The type of border drawn around the progress bar. The available options are
PBBT_NONE	No Border.
PBBT_PLAIN	Plain Black Box (default).
PBBT_RECESSED	Recessed Box.
PBBT_RAISED	Raised Box.
PBBT_RIDGE	Raised Ridge.
PB_TextMode	The mode used for rendering the text. The available options are
PBTM_NONE	No Text (default).
PBTM_PERCENT	Display Value as a percentage, e.g. 50%.
PBTM_VALUE	Display Value as "Value/Size", e.g. 67/153.
PB_TextPosition	The position you want the text to be displayed. The available options are
PBTP_BELOW	Text centred below the bar (default).
PBTP_ABOVE	Text centred above the bar.
PBTP_LEFT	Text to the left of the bar.
PBTP_RIGHT	Text to the right of the bar.
PBTP_CENTRE	Text centred inside the bar.
PB_TextColour	The pen number of the text's foreground colour. The default is 1 (Black).
PB_TextBackColour	The pen number of the text's background colour. The default is 0 (Grey). This value is ignored if the text is centred within the bar. In this case the background will be the bar itself.
PB_TextFont	A pointer to a TextAttr structure defining the font to be used for the text. If this value is NULL the window's font will be used (default).

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() FreeProgBar()

1.4 FreeProgBar()

NAME

FreeProgBar -- Deletes a progress bar from the screen and releases all memory used by it. (V1.0)

SYNOPSIS

```
FreeProgBar ( PBar )
```

```
void FreeProgBar ( PBAR * )
```

FUNCTION

Releases all the memory used by the progress bar and removes it from the screen.

INPUTS

PBar - A pointer to the progress bar to be freed. This is the pointer returned by CreateProgBar(). After calling FreeProgBar() you should not use PBar again.

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() SetProgBarAttrs() ClearProgBar()

1.5 RefreshProgBar()

NAME

RefreshProgBar -- Redraws a progress bar. (V1.0)

SYNOPSIS

```
RefreshProgBar ( PBar )
```

```
void RefreshProgBar ( PBAR * )
```

FUNCTION

Redraws a progress bar in its current state. It does not recalculate any of the progress bar's internal values.

INPUTS

PBar - A pointer to the progress bar to be redrawn. This is the pointer returned by CreateProgBar().

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() SetProgBarAttrs() FreeProgBar() UpdateProgBar()
ResetProgBar()

1.6 UpdateProgBar()

NAME

UpdateProgBar -- Change the current value of a progress bar and redisplay its imagery. (V1.0)

SYNOPSIS

UpdateProgBar (PBar, Value)

void UpdateProgBar (PBAR *, UWORD)

FUNCTION

Updates the current value of the progress bar, performs all the necessary recalculations and redisplay the progress bar.

INPUTS

PBar - A pointer to the progress bar whose value is to be changed. This is the pointer returned by CreateProgBar().
Value - The new value for the progress bar.

NOTES

If the value supplied is greater than the size of the bar (PB_BarSize) the actual value stored will be PB_BarSize. This guarantees that the bar cannot overflow.

If the bar's value is to be changed at the same time as other properties it is better to use SetProgBarAttrs().

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() SetProgBarAttrs() FreeProgBar() RefreshProgBar()
ResetProgBar()

1.7 ResetProgBar()

NAME

ResetProgBar -- Reset the current value of a progress bar to zero and redisplay its imagery. (V1.0)

SYNOPSIS

```
ResetProgBar ( PBar )
```

```
void ResetProgBar ( PBAR * )
```

FUNCTION

Resets the current value of the progress bar to zero, performs all the necessary recalculations and redisplay the progress bar.

INPUTS

PBar - A pointer to the progress bar whose value is to be reset. This is the pointer returned by CreateProgBar().

NOTES

If the bar's value is to be changed at the same time as other properties it is better to use SetProgBarAttrs().

This function is identical to UpdateProgBar(PBar, 0).

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() SetProgBarAttrs() FreeProgBar() RefreshProgBar()
UpdateProgBar()

1.8 ClearProgBar()

NAME

ClearProgBar -- Removes a progress bar from the display. (V1.0)

SYNOPSIS

```
ClearProgBar ( PBar )
```

```
void ClearProgBar ( PBAR * )
```

FUNCTION

Removes the supplied progress bar from the display. It does not release the memory used by the progress bar.

INPUTS

PBar - A pointer to the progress bar whose imagery is to be erased. This is the pointer returned by CreateProgBar().

NOTES

If your progress bar does not have any text associated with it, it will be marginally faster to call ClearBar() instead.

If you only want to delete the progress bar's text then it is better to call `ClearText()`.

RESULT

None

BUGS

None known

SEE ALSO

`CreateProgBar()` `SetProgBarAttrs()` `FreeProgBar()` `ClearBar()`
`ClearText()`

1.9 ClearBar()

NAME

`ClearBar` -- Removes a progress bar from the display, but does not remove any text. (V1.0)

SYNOPSIS

`ClearBar (PBar)`

`void ClearBar (PBAR *)`

FUNCTION

Removes the supplied progress bar from the display, but does not delete the bar's text, if any. It does not release the memory used by the progress bar.

INPUTS

`PBar` - A pointer to the progress bar whose imagery is to be erased. This is the pointer returned by `CreateProgBar()`.

RESULT

None

BUGS

None known

SEE ALSO

`CreateProgBar()` `SetProgBarAttrs()` `FreeProgBar()` `ClearProgBar()`
`ClearText()`

1.10 ClearText()

NAME

`ClearText` -- Removes a progress bar's text from the display. (V1.0)

SYNOPSIS

`ClearText (PBar)`

```
void ClearText ( PBAR * )
```

FUNCTION

Removes the supplied progress bar's text from the display. It does not affect the image of the bar itself.

INPUTS

PBar - A pointer to the progress bar whose text is to be erased. This is the pointer returned by CreateProgBar().

RESULT

None

BUGS

None known

SEE ALSO

CreateProgBar() SetProgBarAttrs() FreeProgBar() ClearProgBar()
ClearBar()