AJUI ProgressBar

User manual

Version Control	Date	Comment (Change)	Author
1.0	06.06.2019	First version	Gary Criblez
1.01	14.06.2019	Subform and progress bar without animation	Gary Criblez

1 Introduction

1.1 What is AJUI ProgressBar

AJUI ProgressBar is a component developed in version 17 R3 of the 4D language. Its use is intended for 4D developers. The advantage of this one is to be able to generate progress bars very easily. It uses SVG and 4D Worker technology to model them in image format and bring them to life.

The component creates instances. Instances are objects composed of a set of properties that will define the characteristics and visual rendering of the progress bars. They also have formulas to access assessors and methods for generating progress bars.

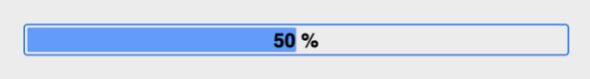
The idea behind the implementation of AJUI ProgressBar is to provide a visual tool that allows users to get an overview of the progress of a process by displaying a progress bar.

In particular, the component makes it possible to inform the user about the progress of an ongoing process by providing visual and valued indicators.

1.2 The progress bars

The component contains two types of instance representing two specific forms of progress bar that are made available. The first being linear in shape and the second circular. Each one has properties specific to their design.

Linear bar :



Circular bar :

The numerical values that can be assigned to the bars range from 0 to 100, but since the text displayed can be customized, it is possible to visually represent other values such as steps.

The component also uses Worker to refresh the page when generating progress bars. However, the Workers have another utility that gives us the ability to manage animations. The animations are applicable to both types of progress bar.

The first applicable animation is a transition effect between the old and the new value you have assigned to your progress bar. By determining a transition speed, the component will update the image form object and refresh the form a certain number of times until it reaches the new value.

The second animation is what is called a "Barbershop". In this case, the progress bar no longer relies on a fixed value and rotates in a loop (circular bar) or performs back and forth (linear bar) until the developer decides to stop it by using the corresponding stop formula. This type of animation is perfectly applicable to a process whose processing time is unknown. The purpose of this animation is to make the user wait by informing him that the process is in progress.

1.3 Restriction with subforms

In the current version, animations and the "barbershop" using the Worker are not usable for subforms because we use a "Call Form" that goes back to the first level of the form.

The use of the linear and circular bar without animation mode does not present any problem for their use in the context of subforms.

2 Component methods

The component provides two methods to retrieve instances for each type of progress bar.

New AJUI_ProgressBar : the method returns an object representing an instance of the linear progress bar. The object contains different properties and their default values as well as formulas to manipulate them.

New AJUI CircularProgressBar : the method returns an object representing an instance of the circular progress bar. The object contains different properties and their default values as well as formulas to manipulate them.

The other methods available prefixed "HDI" (How do I) allow you to call forms containing examples of how to use the component's features.

3 List of properties

Each instance has a set of properties that will allow you to define its representation in the host form. In this chapter, we will review the different existing properties accessible by a formula that acts as Setter but also as Getter if no parameters are passed to them. All formulas can be called up at the first level of the object.

3.1 Properties of the linear progress bar

3.1.1 Progress Area

Properties related to the progress area on the second level of the object : MyBar.**progressArea**

Name	Type	Description	Default value	Formula
backgroundColor	string	Font color of the background	none	PBBGColor
borderColor	string	Color of the outer border	#3276DC	PBBorderColor
borderSize	longint	Size of the outer border	1	PBBorderSize
cornerRadius	longint	Corner radius of the outer border	2	PBCornerRadius
Height	longint	Height of the progress area	20	PBAreaHeight
Width	longint	Width of the progress area	360	PBAreaWidth

Formulas and their parameters

Formula name	Parameter(s)
PBBGColor (color)	- Font color (string)
PBBorderColor (color)	- Border color (string)
PBBorderSize (size)	- Border size (longint)
PBCornerRadius (radius)	- Corner radius (longint)
PBAreaHeight (height)	- Height (longint)
PBAreaWidth (widht)	- Width (longint)

3.1.2 SVG Area

Properties related to the SVG area on the second level of the object : MyBanner.**svgArea**

Name	Type	Description	Default value	Formula
windowBGColor	string	Background color of the SVG Area. It is possible to define an opacity rate (%). See default value.	white:0	PBWindowColor

Formulas and their parameters

Formula name	Parameter(s)
PBWindowColor (color)	- Background color (string)

3.1.3 Textual content

Properties related to textual content at the second level of the object : MyBanner.**text**

Name	Type	Description	Default value	Formula
position	string	Position of the title in relation to the progress bar. Possible values: "top", "middle" and "bottom". Note that the font size of the title adapts to the size of the bar if it is positioned inside (middle).	middle	PBTextPosition
title	string	Title label. Two placeholders are available to display the current value of the progress bar. Placeholders : - Value in longint : $value_l - Value in real : $value_r	$value_l	PBTitle

Formulas and their parameters

Formula name	Parameter(s)
PBTextPosition (position)	- Title position (string)
PBTitle (label)	- Title label (string)

3.1.4 Progress bar

Properties related to the bar on the second level of the object : MyBanner.**bar**

Name	Type	Description	Default value	Formula
color	string	Color of the progress bar. It is possible to define an opacity rate (%). See default value.	#3B88FD:80	PBBarColor
offset	real	Padding between the outer border and the progress bar.	1.5	PBBarOffset
pictureFormObjName	string	Name of the image form object hosting the progress bar.	empty string	PBPictFormObjName
speed	real	Speed represents tics. This corresponds to the transition time between each percent during the animation.	1	PBBarSpeed

Formulas and their parameters

Formula name	Parameter(s)
PBBarColor (color)	- Bar color (string)
PBBarOffset (size)	- Padding size (real)
PBPictFormObjName (name)	- Name of the form object (string)
PBBarSpeed (speed)	- Animation speed (real)

3.1.5 Percentage and format of the text

Properties related to the percentage and format of the text on the second level of the object : MyBanner.**percent**

Name	Type	Description	Default value	Formula
animation	boolean	Enable/disable animation effects on transitions between the old and new values.	False	PBActivateAnimation
display	boolean	Show/hide title	True	PBDisplayText
font.color	string	Font color applied to text elements	black	PBFontColor
font.name	string	Font name applied to text elements	Helvetica, Arial	PBFontName
font.size	longint	Font size applied to text elements	12	PBFontSize
font.style	string	Style applied to text elements. The following values are accepted : - Bold - Italic - Strikethrough	bold	PBFontStyle

		- Underline		
value	real	Current value of the progress bar. The value can range from 0 to 100.	0	PBPercentValue

Formulas and their parameters

Formula name	Parameter(s)
PBActivateAnimation (activation)	- Activation of animations (boolean)
PBDisplayText (show)	- Show/hide text (boolean)
PBFontColor (color)	- Font color (string)
PBFontName (font)	- Font name (string)
PBFontSize (size)	- Font size (longint)
PBFontStyle (style)	- Font style (string)
PBPercentValue (value)	- Current value (real)

3.2 Properties of the circular progress bar

3.2.1 Size of the circle

Property related to the size of the circle on the second level of the object : MyBar.**cpb**

Name	Type	Description	Default value	Formula
radius	longint	Size of the radius of the progress bar (pix)	100	CPBExternalRadius
autoRadiusSize	boolean	Enables automatic calculation of the radius size according to the size of the form object associated with the instance. The smallest value between the width and length of the form object is used.	True	CPBRadiusAutoSize

Formulas and their parameters

Formula name	Parameter(s)
CPBExternalRadius (radius)	- Radius of the bar (longint)
CPBRadiusAutoSize (activate)	- Activate auto calculation (boolean)

3.2.2 Bar background

Properties related to the background of the bar on the third level of the object : MyBanner.cpb.**bgBar**

Name	Type	Description	Default value	Formula

color	string	Background color of the background of the progress bar.	lightgray	CPBBackgroundColor
fillColor	string	Background color of the center.	none	CPBBackgroundFillColor
width	longint	Width of the bar background	16	CPBBackgroundWidth

Formulas and their parameters

Formula name	Parameter(s)
CPBBackgroundColor (color)	- Background color (string)
CPBBackgroundFillColor (color)	- Center color (string)
CPBBackgroundWidth (size)	- Background size (longint)

3.2.3 Progress bar

Properties related to the progress bar on the third level of the object : MyBanner.cpb.**bar**

Name	Type	Description	Default value	Formula
color	string	Color of the progress bar. It is possible to define an opacity rate (%). See default value.	#3B88FD:80	CPBBarColor
linecap	string	The shape of the extremity of the progress bar. Possible values : - butt - round - square	butt	CPBBarLinecap
pictureFormObjName	string	Name of the image form object hosting the progress bar.	empty string	CPBPictFormObjName
speed	real	Speed represents tics. This corresponds to the transition time between each percent during the animation.	1	CPBBarSpeed
width	longint	Width of the progress bar.	12	CPBBarWidth

Formulas and their parameters

Formula name	Parameter(s)
CPBBarColor (color)	- Bar color (string)
CPBBarLinecap (shape)	- Shape of the extremity (string)
CPBPictFormObjName (name)	- Name of the form object (string)
CPBBarSpeed (speed)	- Animation speed (real)
CPBBarWidth (width)	- Bar width (string)

3.2.4 Textual content

Properties related to text content on the third level of the object : MyBanner.cpb.**text**

Name	Type	Description	Default value	Formula
subtitle	string	Subtitle label. Two placeholders are available to display the current value of the progress bar. Placeholders : - Value in longint : $value_l - Value in real : $value_r	%	CPBSubTitle
title	string	Title label. It uses the same placeholders in relation to the subtitles.	$value_l	CPBTitle

Formulas and their parameters

Formula name	Parameter(s)
CPBSubTitle (label)	- Subtitle label (string)
CPBTitle (label)	- Title label (string)

3.2.5 Percentage and text format

Properties related to the percentage and format of the text on the third level of the object : MyBanner.cpb.**percent**

Name	Type	Description	Default value	Formula
animation	boolean	Enable/disable animation effects on transitions between the old and new values.	False	CPBActivateAnimation
display	boolean	Show/hide title	True	CPBDisplayText
font.color	string	Font color applied to the text elements. It is possible to define an opacity rate (%). See default value.	gray :80	CPBFontColor
font.name	string	Font name applied to the text elements.	Helvetica, Arial	CPBFontName
font.size	longint	Font size applied to the text elements.	35	CPBFontSize
font.style	string	Font style applied to text elements. The following values are accepted: - Bold - Italic - Strikethrough - Underline	Bold	CPBFontStyle

value	real	Current value of the progress bar. The value can range from 0 to 100.	0	CPBPercentValue

Formulas and their parameters

Formula name	Parameter(s)
CPBActivateAnimation (activation)	- Activation of animations (boolean)
CPBDisplayText (show)	- Show/hide text (boolean)
CPBFontColor (color)	- Font color (string)
CPBFontName (font)	- Font name (string)
CPBFontSize (size)	- Font size (longint)
CPBFontStyle (style)	- Font style (string)
CPBPercentValue (value)	- Current value (real)

4 Formulas to manage the progress bar

The following Formulas are not directly linked to a property but will use them to manage the life cycle of the progress bar.

4.1 Linear progress bar

- **LaunchProgressBar** : Generates the progress bar and associates it with the corresponding image form object. Manages the transition animation if enabled.
- **LaunchBarberShop** : Generates the progress bar and associates it with the corresponding image form object. Manages the "barbershop" animation until it is interrupted.
- **StopPBProcess** : Cancel animations and the Worker corresponding to the instance.

4.2 Circular progress bar

- **LaunchCircularProgressBar** : Generates the progress bar and associates it with the corresponding image form object. Manages the transition animation if enabled.
- **LaunchCircularBarberShop** : Generates the progress bar and associates it with the corresponding image form object. Manages the "barbershop" animation until it is interrupted.
- **StopCPBProcess** : Cancel animations and the Worker corresponding to the instance.

Be careful, the image form object is automatically resized after the SVG generation regardless of the type of bar. However, its position in X and Y does not change.

5 Worker and storage 4D

The component uses Workers and 4D Storage to manage the generation and animation of progress bars.

Each instance will use its own Worker. The name of this one is prefixed "AJUI" and includes the property "pictureFormObjName". However, it is possible to use a single Worker for several instances under three conditions.

- Use the same image form object and so each instance receives the same name for the "pictureFormObjName" property.
- Is of the same type of bar (linear or circular).
- Not used simultaneously (the first condition should already enforce this one).

Concerning storage, it should be noted that each time you want to launch the barbershop animation, the component will store in the object: storage.barbershop.run the name "pictureFormObjName" as a property.

This property will be of the Boolean type and will indicate to the component when to execute and interrupt the "barbershop" animation for the instance concerned. In concrete terms, it acts as a flag. So be careful if you also use 4D storage in your application to not delete this object.

6 Getting started

This chapter is used to introduce you to how to easily set up a progress bar.

The steps that will be discussed are applicable for both types of progress bar.

We will use the default values of the properties for this example. You are free to consult the list of formulas if you wish to customize your bar during this presentation.

The first step is to add an image object to a form. This one will be used to host the progress bar. You can add other image objects if you want to generate several progress bars simultaneously.

The second step requires to create an instance, you can choose to launch one of the two methods to create an instance of the component (circular or linear bar). Ideally, the object returned by the method of your choice should be stored in the "Form" variable, it is also recommended to add a level to the "Form" object. For example, "Form.progressbar".

Once this is done, add the name of your image form object to the instance. Then for your test, we propose you to modify the value of the current progression which is by default 0 and to activate the animations. All you have to do is launch the corresponding "launch" formula to generate your progress bar.

Repeats the second step for each image form object if you have created several (one instance per image object).

Also tried to use the formula to activate the "barbershop" animation. Don't forget to create a button to activate the formula that stops this animation (see the chapter : Formulas to manage the progress bar).

Sample from the code for a linear progress bar :

```
Form.mypb:=New object
Form.mypb:=New AJUI_ProgressBar

Form.mypb.PBActivateAnimation(True)
Form.mypb.PBPercentValue(60)
Form.mypb.PBPictFormObjName("AJUI_BPPict_container")

Form.mypb.LaunchProgressBar()
```

We also advise you to launch the example forms with the prefixed methods "HDI" of the component to get an idea of what it looks like visually and discover some use cases where AJUI ProgressBar brings you a real added value.

7 Conclusion

The purpose of this document was to introduce you to the theoretical principles of the component as well as the different methods, formulas and properties at your disposal to be able to manage the generation of progress bars.

As for the practical elements presented, they are intended to allow you to get off to a good start and to address some specific cases that could arise in the use of the component.

You want help for the implementation of the component AJUI_ProgressBar in your application. You want to modify or extend its functionalities for a specific purpose. You want to have the source code of the AJUI_ ProgressBar component in order to perennize its use in your application with future versions of 4D. Feel free to contact us to discuss it.