EB GUIDE tutorial

Creating a horizontal progress bar

Version 6.11.0.210526170413

1. Tutorial: Creating a horizontal progress bar

TIP

Default window layout

All instructions and screenshots use the default window layout. If you want to follow the instructions, we recommend to set the EB GUIDE Studio or EB GUIDE Monitor window to default layout by selecting **Layout** > **Reset to default layout**.

The following instructions guide you through the process of modeling a progress bar as shown below.

Figure 1. Progress bar

You can also have a look at the progress bar template in the widget template library. See https://www.elektrobit.com/ebguide/examples/.

Approximate duration: 15 minutes

Adding the widgets

The following instructions guide you through the process of adding widgets for the progress bar.

Prerequisite:

- The **Main** state machine contains an Initial state and a View state.
- The Initial state has a transition to the View state.
- The content area displays a View.

Step 1
In the **Templates** component, click + and then select `Container`.

A template is created that contains a Container.

Step 2
Rename the template to `T_ProgressBar`.

Step 3
Rename the Container to `ProgressBar_Container`.

Step 4
Drag a Rectangle into the Container and rename it to `Background_Rectangle`.

Step 5
Drag another Rectangle into the Container and rename it to `Progress_Rectangle`.

This Rectangle visualizes the progress of the operation.

Step 6
Drag a Label into the Container and rename it to `Percentage_Text`.

Entering the properties for the progress bar

The following instructions guide you through the process of configuring the properties and adding scripts to the widgets.

Prerequisite:

- You completed the previous instruction.

Step 1
In the **Templates** component, select `ProgressBar_Container`.

Step 2
Add the properties `width`, `height`, `x`, `y` to the template interface.

To add a property to the template interface, in the **Properties** component, click the button next to the property. In the menu, click **Add to template interface**. The icon is displayed next to the property.

Step 3
In the **Properties** component, go to the **User-defined properties** category and click + and select `Integer`.

A user-defined property of type `Integer` is added to the Container.

Step 4
Rename the property to `progress`.

Step 5
Add `progress` to the template interface.

Step 6
In the **Templates** component, select `Background_Rectangle`.

Step 7
Link the `width` to the width property of `ProgressBar_Container`.

To link a property to another property, in **Properties** component, click the button next to the property. In the menu, click **Add link to widget property**.

A dialog opens.

Step 8
In the dialog, select the `width` property of `ProgressBar_Container` and click **Accept**.

Step 9
Link the `height` property of `Background_Rectangle` to the height property of `ProgressBar_Container`.

Step 10
In the **Templates** component, select `Progress_Rectangle`.

Step 11
Link the `height` property to the `height` property of `ProgressBar_Container`.

Step 12
Set the `fillColor` to green.

Step 13
Next to the `width` property click  and then select `Convert to script`.

The `width` property defines the width as a percentage of the width of `ProgressBar_Container`.

Step 14
Click {}.

An EB GUIDE Script editor opens.

Step 15
Enter the following EB GUIDE Script in the **Read** section:

```
function()
{
 v:this->^.width * v:this->^.progress / 100
}
```

This script divides the value of the `progress` property by 100.

Step 16
Click **Add available triggers to list.**

Two triggers for `width` and `progress` are added.

Step 17
In the **Templates** component, select `Percentage_Text`.

Step 18
Link the `width` and `height` properties to the width and height of `ProgressBar_Container`.

Step 19
Set the `horizontalAlign` to `center (1)`.

Step 20
Convert the `text` property into a script.

The text will display the percentage of the width of the container.

Step 21
Click {}.

An EB GUIDE Script editor opens.

Step 22
In the **Read** section, enter the following script:

```
function()
```

```
{
 f:int2string(v:this->^.progress) + "%"
}
```

This script converts the percentage value into a string and adds the % character after the percentage number.

Step 23
Click **Add available triggers to list**.

The trigger for `progress` is added.

Step 24
Set the `x` and `y` properties of all widgets in the template to 0.

Step 25
In the **Navigation** component, double-click the View.

Step 26
From the **Toolbox** component, drag `T_ProgressBar` into the content area.

The template is added to the View. Now you can add an Animation to it to show the dynamic progress of an operation.

Animating the progress

The following instructions guide you through the process of animating the progress bar, so that you can see better what happens when you change the percentage value.

Prerequisite:

- You completed the previous instruction.
- The content area displays the View.

Step 1
Drag an Animation into the View.

Step 2
Rename the Animation to `Loading_Animation`.

Step 3
In the **Properties** component, go to the **User-defined properties** category, click `+`, and select `Conditional script`.

Step 4
Rename the conditional script to `animateProgress`.

Step 5
Next to the conditional script property click `{}`.

An EB GUIDE Script editor opens.

Step 6
In the **On trigger** section, enter the following script:

```
function(v:arg0::bool)
{
 f:animation_play(v:this)
 false
}
```

Step 7
In the **Navigation** component, double-click `Loading_Animation` to open the **Animation editor**.

Step 8
In the **Animation editor**, next to **Animated properties** click + and select `View 1`.

The **Animation properties** dialog opens.

Step 9
Under `T_ProgressBar 1`, select the `progress` property and then `Linear interpolation curve`. Click **Accept**.

Step 10
In the **Properties** component, set `end` property to `100`.

The progress animation will stop when the progress indicator reaches 100%.

Saving and testing the EB GUIDE model

Prerequisite:

- You completed the previous instruction.

Step 1
To save the project, click in the command area.

Step 2
To start the simulation, click in the command area.