EB GUIDE tutorial

Adding a language-dependent text to a datapool item

Version 6.7.1.149092

Copyright © 2018 Elektrobit Automotive GmbH

Legal notice

1. Tutorial: Adding a language-dependent text to a datapool item

NOTE	**Default window layout**
(i)	All instructions and screenshots of this user's guide use the default window layout. If you want to follow the instructions, we recommend to set the EB GUIDE Studio window to default layout by selecting **Layout** > **Reset to default layout**.

EB GUIDE offers the possibility to display texts in the user's preferred language. The following instructions show you how to model a label that changes with an English, French, and German user interface.

Approximate duration: 15 minutes

NOTE	**Prerequisites to language dependency**
(i)	To add language support to a datapool item, do the following:

    ► If its `Value` property is linked to another datapool item or widget property, remove the link.

    ► If its `Value` property is a scripted value, convert the property to a plain value.

👣   Linking a widget property to a datapool item

The following instructions guide you through the process of linking the label's `text` property to a datapool item. In run-time the displayed text is provided by the datapool item.

Prerequisite:

- Three languages are added to the EB GUIDE model: English, German, and French.The name of **Language 1** is set to `German` and the name of **Language 2** is set to `French`.
- The content area displays the **Main** state machine.
- The Main state machine contains an initial state and a view state.
- The content area displays the view.
- The initial state has a transition to the view state.
- The view state contains a label.

Step 1
Click the label.

Step 2
In the **Properties** component, go to the `text` property, and click the ☐ button next to the property.

<u>Step 3</u>
In the menu, click **Add link to datapool item**.

A dialog opens.

<u>Step 4</u>
To add a new datapool item, enter `Welcome_text` in the text box.

<u>Step 5</u>
Click **Add datapool item**.

<u>Step 6</u>
Click **Accept**.

The datapool item `Welcome_text` is added.

In the content area, the label no longer displays any text.

                    Enter language-dependent text to the datapool item

The following instructions guide you through the process of adding language-dependent text to the datapool item. For every language the `Value` property has a different text.

Prerequisite:

- You completed the previous instruction.

<u>Step 1</u>
In the **Datapool** component, click the `Welcome_text` datapool item.

<u>Step 2</u>
Click the ☐ button.

<u>Step 3</u>
In the menu, click **Add language support**.

In the **Properties** component, the language properties are displayed.

<u>Step 4</u>
In the **Datapool** component, in the `Value` text box, enter `Welcome`.

In the content area, the label displays `Welcome`.

<u>Step 5</u>
Go to the **Properties** component.

<u>Step 6</u>
In the `German` text box, enter `Willkommen`.

In the `Language` box in the upper left corner, change the language to `German`.

In the content area, the label displays `Willkommen`.

Step 7

In the `French` text box, enter `Bienvenue`.

In the `Language` box in the upper left corner, change the language to `French`.

In the content area, the label displays `Bienvenue`.

You have added language support for English, German and French and defined a language-dependent text label.

Changing the language during run-time

The following instructions guide you through the process of creating a script for changing the language during run-time. Each time, the user clicks the label, the display language changes.

Prerequisite:

- You completed the previous instruction.

Step 1

In the **Datapool** component, click ➕.

A menu expands.

Step 2

In the menu, click `Integer`.

A datapool item of type `Integer` is added.

Step 3

Rename the datapool item to `SelectedLanguage`.

Step 4

In the **Navigation** component, click the `Label 1` label.

Step 5

In the **Properties** component, go to the **Widget feature properties** and click **Add/Remove**.

The **Widget features** dialog is displayed.

Step 6

Under **Available widget features**, expand the **Input handling** category, and select the **Touch pressed** widget feature.

Step 7

Click **Accept**.

The related widget feature properties are added to the label and displayed in the **Properties** component.

Step 8

Next to the `touchPressed` property, click **Edit**.

<u>Step 9</u>
Replace the existing EB GUIDE Script with the following code:

```
function(v:touchId::int, v:x::int, v:y::int, v:fingerId::int)
{
    if (dp:SelectedLanguage == 0)  // Standard selected
    {
     f:language(l:German)
     dp:SelectedLanguage = 1
    }
    else if (dp:SelectedLanguage == 1)  // German selected
    {
     f:language(l:French)
     dp:SelectedLanguage = 2
    }
    else if (dp:SelectedLanguage == 2)  // French selected
    {
     f:language(l:Standard)
     dp:SelectedLanguage = 0
    }
    false
}
```

<u>Step 10</u>
Click **Accept**.

You configured the label and wrote an EB GUIDE Script which changes the language of the label during run-
time.

Result:

You added a datapool item of type `String` to the EB GUIDE model. The datapool item has different values
for languages. In English the value is `Welcome`. In German the value is `Willkommen`. In French the value is
`Bienvenue`. The datapool item is linked to the `text` property of the label. Every time you change the language
of the EB GUIDE model the text of the label changes too.