

The Many Faces of F1

Siu Chong Wan and Sheba Ephraim, Westat

Presenter: Siu Chong Wan

1. Abstract

The display of help text to interviewers or respondents, depending on the mode of collection, is an essential feature in Blaise. While we used function keys (F1) for online help in Blaise 4, we have additional options in Blaise 5.

In addition, we can use the same techniques to display other materials, like showcards, that we often need in interviewer-administered surveys.

However, what we are going to explore here is not limited to interviewer-administered surveys.

This paper discusses our use of different triggers in Blaise 5 and the types of actions that support the needs of displaying supplementary survey materials.

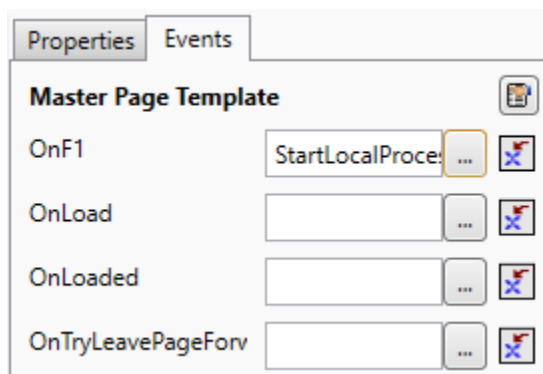
2. Triggers

There are a variety of ways for the users to access the help text. While traditionally trained interviewers might still favor keyboard shortcuts, the growing availability of touchscreen devices makes buttons and hyperlinks more attractive. We will discuss the triggers that we have tried so far.

2.1 Keyboard Shortcuts

Be it the F1 key again, or any other function key or shortcut key, we can use template events to define actions that display help text.

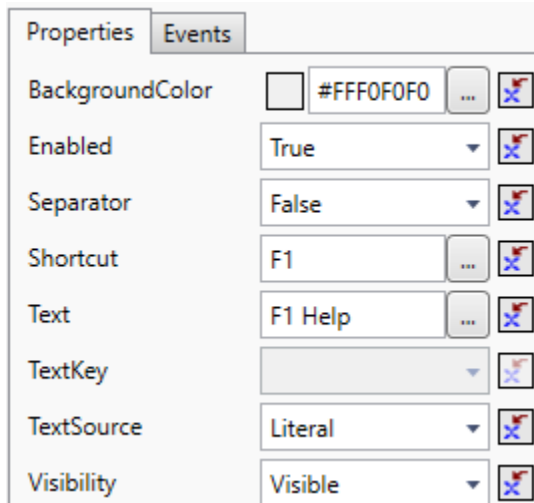
This is an example of the “OnF1” event defined in a Master Page Template.



2.2 Menu Items

Using a menu control, we can create menu item events to define actions that display help text. In addition, we can use the shortcut property in a menu item to define an accelerator key that works just like the abovementioned keyboard shortcuts.

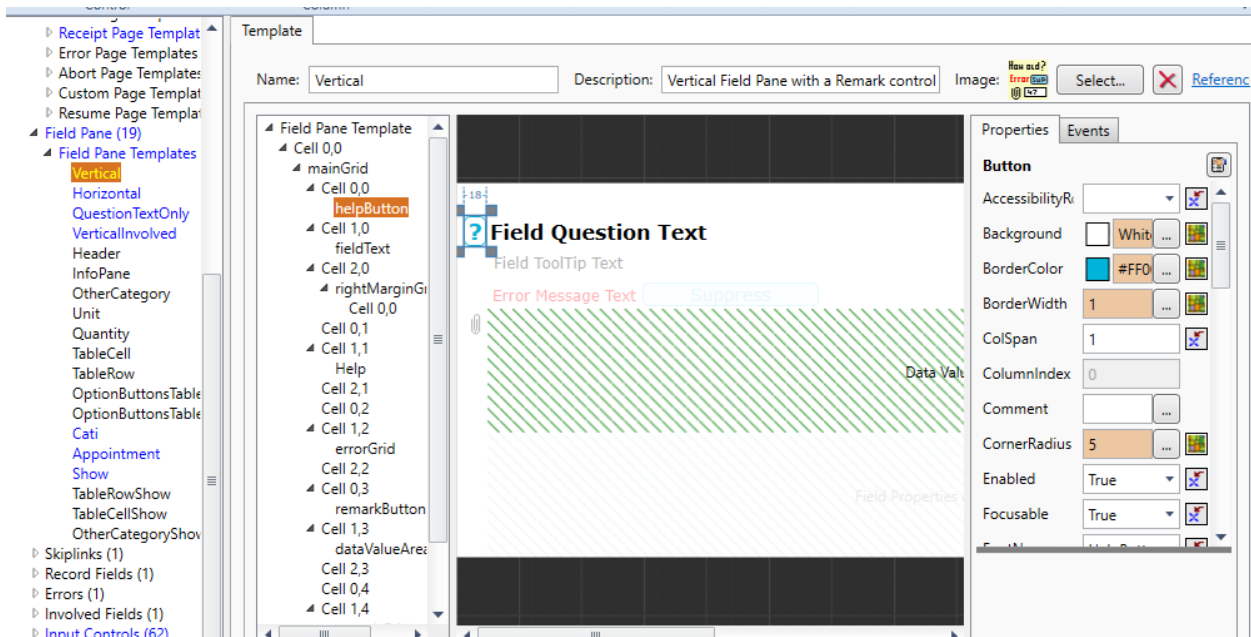
This is an example of the F1 key defined in the shortcut property of a menu item.



2.3 Buttons

Buttons might be one of the most versatile ways to show help text. Depending on what kind of template the button resides in, we can display help text associated with a page, a field, a response category, or even a table column header.

This is an example of a button shown as a blue “?” that triggers help text in a Field Pane Template.



2.4 Hyperlink Tag

The hyperlink tag supports the OnClick attribute, where we can define actions that display help text. This can be defined in the field text alone in the FIELDS section of the data model without defining anything in the resource database.

This is an example of a hyperlink defined in the question text.

```
Q2 (Q2)
"This is an example for online help.<right>
<hyperlink Name="helplink" Text="HELP"
  OnClick="{Action
    StartLocalProcess('C:\\Windows\\hh.exe',
      'mk:@MSITStore:C:\\Help\\Help.chm::/RespondentHelp.htm',
      ', True, Normal) }\"
>Help</hyperlink></right>"
: (OK)
```

3. Actions

There are different actions to display extra reading materials. Here, we focus on ways to open local files, web pages, and simply extra field text in the current page. While local files can be accessed by the StartLocalProcess action, web pages can be accessed by the GotoUri action. If we want to remain on the current page the whole time, we simply hide and show field text.

3.1 StartLocalProcess Action

The StartLocalProcess action introduced in Blaise 5.13 provides a lot of possibility in Blaise 5. As long as we can find a local program that can display additional materials, we can use it. Here are a few examples:

To display HTML-Help with topic identifier:

```
{ Action StartLocalProcess('C:\\Windows\\hh.exe',
'mk:@MSITStore:C:\\Documents\\Help.chm::/RespondentHelp.htm', ", True, Normal) }
```

To display a JPEG file in MS Edge:

```
{ Action StartLocalProcess('C:\\Program Files (x86)\\Microsoft\\Edge\\Application\\msedge.exe', '--
app="C:\\Documents\\English_2022_PP_16_593x730.jpg"; ', ", True, Normal) }
```

To display a PDF file in MS Edge at a particular page:

```
{ Action StartLocalProcess('C:\\Program Files (x86)\\Microsoft\\Edge\\Application\\msedge.exe', '--
app="C:\\Documents\\Help\\LayoutDesigner_Tutorial_PartI.pdf#page=5"; ', ", True,
Normal) }
```

3.2 GotoUri Action

The GotoUri action is definitely the way to go when we need to access help text on a web page. As long as the respondent has access to the internet, we can display anything from a web source. On the other hand, it can also display local files. Here are a couple examples:

To go to a web page:

```
{ Action GotoUri('https://www.westat.com/about-westat/') }
```

To go to a local file:

```
{Action GotoUri('C:\\Documents\\Help\\HTM\\RespondentHelp.htm')}
```

The advantage of maintaining the material in a web source is that the updates to the web source can be available immediately. This is particularly good for time-sensitive information. However, the requirement of internet access could also be a weakness because we do not always have internet access in interviewer-administered surveys. In other words, the lack of internet access is almost assumed in interviewer-administered surveys.

3.3 ToggleVisibility Action

One thing we often try to avoid in web surveys is popups. Since we cannot control web respondents' browsers and cannot disable popup blockers, we usually try to display help text in the current page. In this case, we do not start any local process or go to any URI. We simply create a text role for the help text and use the ToggleVisibility action to show and hide the role text.

The predefined text roles "Help" and "ToolTip" are handy for this purpose, but additional text roles can be defined for additional text displaying needs.

In the default Blaise 5 resource database, a "Help" text role is already defined to work with the "helpButton" in the Vertical Field Pane Template to show/hide the help text.

```
<TextRole>
  <Name>Help</Name>
  <Description>A comprehensive instruction for the respondent</Description>
  <TextRoleType>UserDefined</TextRoleType>
</TextRole>
```

```
<Button Name="helpButton" ... OnClick="{Action
ToggleVisibility('Help')} ">
```

4. Examples

We will discuss in more detail how things work in a few examples.

4.1 StartLocalProcess Action Working with a Menu Bar Item

We first define the text role "HelpFile" in both the resource database and the data model for the name of the file that stores the help text associated with the field. This will serve as the topic identifier in HTML-Help.

```
<TextRole>
  <Name>HelpFile</Name>
  <Description>The help text file name.</Description>
  <TextRoleType>UserDefined</TextRoleType>
</TextRole>
```

```
Q1 (Q1)
"This is an example of showing help text with Menu (F1 Help). Please click the menu bar item or press F1.<br>"
HelpFile "LIVEUSHelp"
: (OK)
```

Then we define a menu item in the Master Page Template.

```
<MenuItem Text="Help F1" Shortcut="F1" Visibility="{Expression IF
Page.ActiveField.RoleTextExists('HelpFile') = True THEN 'Visible' ELSE
'Collapsed' ENDIF}" OnClick="{Action Conditional({Expression
Page.ActiveField.RoleTextExists('HelpFile') = True},{Action
StartLocalProcess('C:\\Windows\\hh.exe',{Expression
'mk:@MSITStore:C:\\Documents\\Help.chm:/' +
Page.ActiveField.GetRoleText('HelpFile') +
'.htm'},',',True,Normal)},',')}" />
```

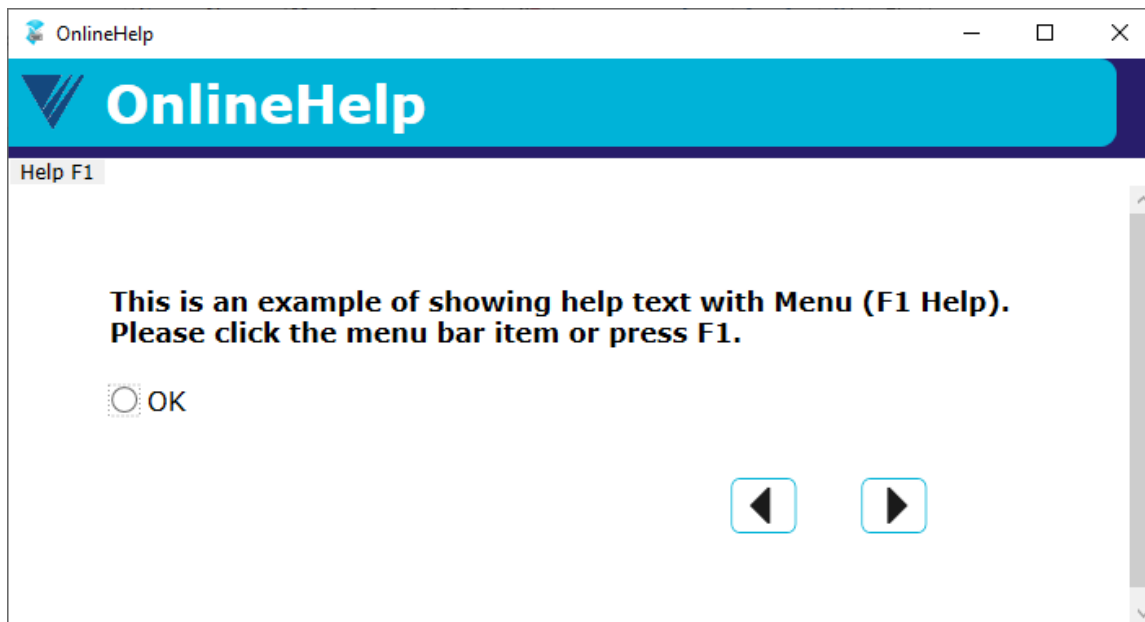
We use “F1” in its shortcut property so that both the menu bar and the F1 key work the same way.

We control its visibility based on whether the active field has “HelpFile” role text defined.

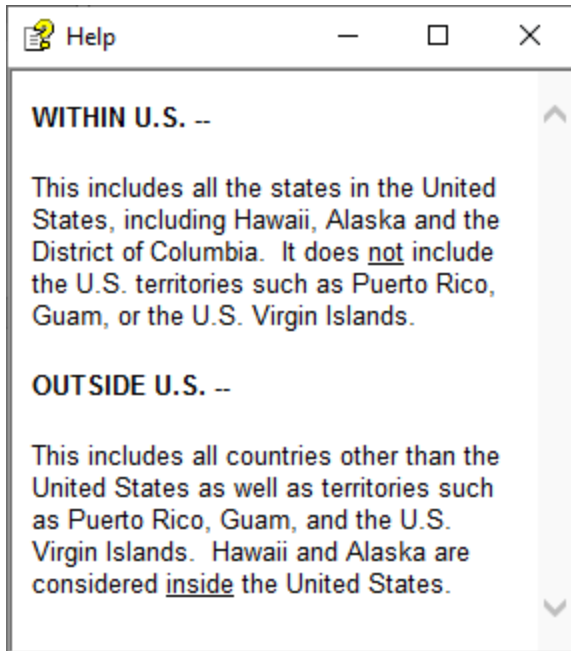
In the OnClick event, we use the StartLocalProcess action to run the local program “hh.exe” to open the HTML-Help file, “Help.chm,” at the page defined in the “HelpFile” role text.

In this example, we define the HelpFile role text as “LIVEUSHelp” so that the StartLocalProcess will run the local hh.exe program to open the topic identifier “LIVEUSHelp” in the C:\Documents\ Help.chm file.

As a result, the Q1 page shows “Help F1” on the menu bar.



By either clicking the menu bar item “Help F1” or pressing the F1 key, the help window pops up with the definition of living within or outside the U.S.



4.2 StartLocalProcess Action Working with a Keyboard Shortcut

Continuing with what we built above, we add the text role “HelpPage” for the page we want to show in a PDF file.

In this example, we use page number 30.

```
Q3 (Q3)
"This is an example of showing help text in a certain page of a PDF file in Edge with the F3 key. Please press F3.<br>"
HelpPage "30"
: (OK)
```

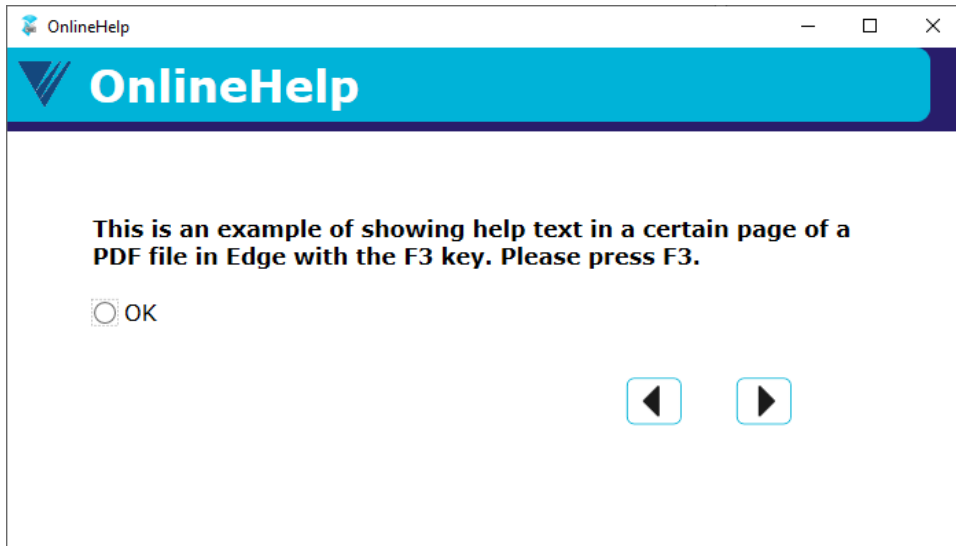
We add the shortcut key F3 to the Vertical Field Pane Template.

```
<Shortcut Key="F3" OnExecute="{Action Conditional({Expression
Field.RoleTextExists('HelpPage') = True},{Action
StartLocalProcess('C:\\Program Files
(x86)\\Microsoft\\Edge\\Application\\msedge.exe',{Expression '--
app=&quot;C:\\Documents\\Help\\Blaise.pdf#page=' +
Page.ActiveField.GetRoleText('HelpPage') +
'&quot;'},',',True,Normal)},',')}" />
```

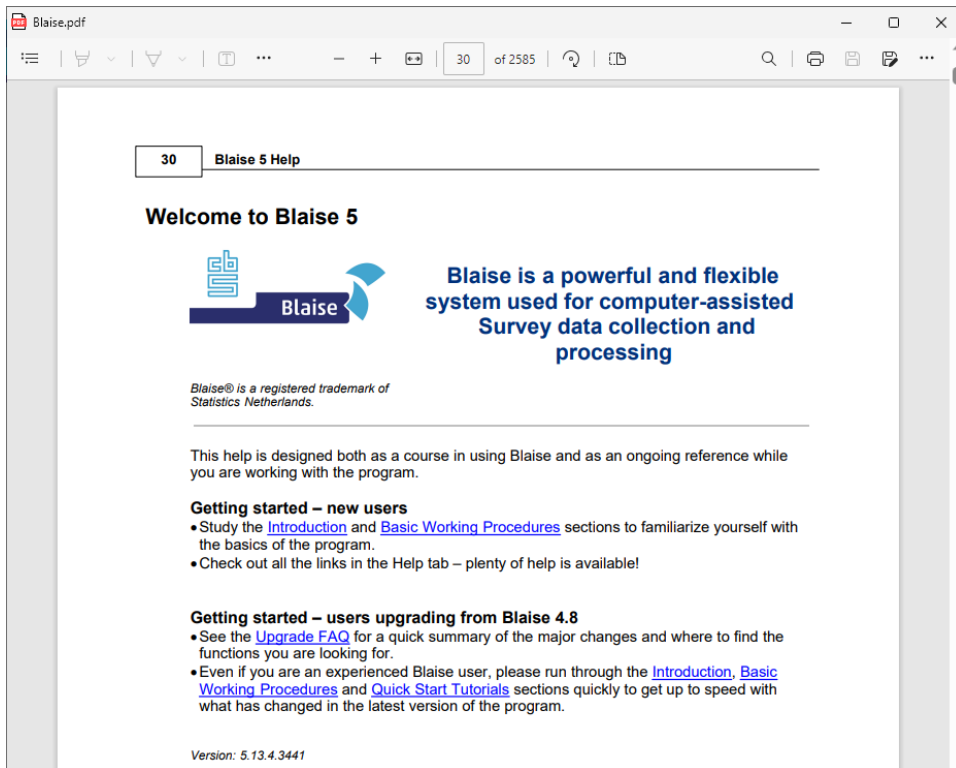
This time, in the OnF3 event, we use the StartLocalProcess action to run the local program “msedge.exe” to open a PDF file in MS Edge at the page defined in the “HelpPage” role text.

In this example, we are showing the Blaise 5 Manual file, Blaise.pdf.

As a result, this page does not need to display any buttons or the menu bar.



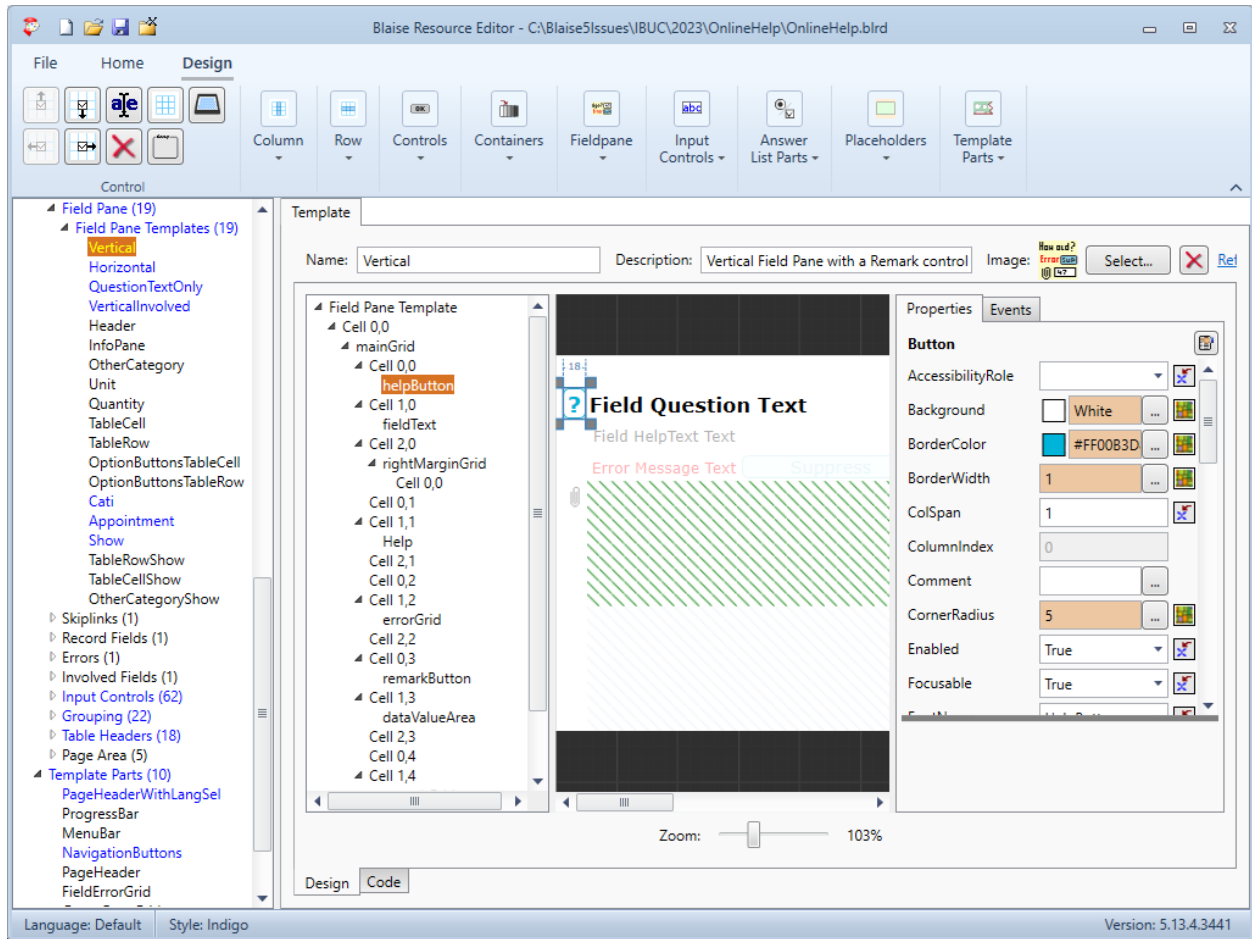
When we press F3, MS Edge pops up to display page 30 of the Blaise 5 Manual.



4.3 Button to Toggle Role Text Visibility

In web surveys, where we cannot start a local process and in general avoid popup windows, role text fits the purpose of showing help text. We usually use a button to toggle the visibility of the role text so that the user can choose to show or hide the role text.

In the resource database, we define a “helpButton” in the Field Pane Template where its visibility is determined by whether the field’s role text “HelpText” exists. Its OnClick event simply shows or hides the HelpText.

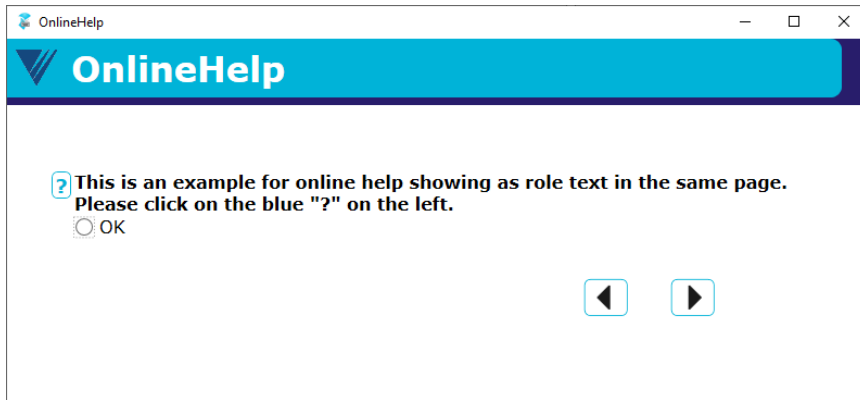


```
<Button Name="helpButton" Visibility="{Expression IF
LEN(Field.GetRoleText('HelpText')) = 0 THEN 'Hidden' ELSE 'Visible'
ENDIF}" HorizontalAlignment="Center" Background="{Style
HelpButtonBackGround}" Margin="{Style HelpButtonMargin}" Width="{Style
HelpButtonWidth}" Height="25" ScreenReaderText="Help"
ToolTipFontName="ToolTip" OnMouseDown="{Action
SetControlProperty('helpButton', 'Background', 'Brush', '{Style
ButtonBackgroundSelected})}" OnMouseUp="{Action
SetControlProperty('helpButton', 'Background', 'Brush', '{Style
HelpButtonBackGround}')}" Text="?" FontName="HelpButton"
ScreenReaderTextSource="Literal" BorderWidth="{Style
HelpButtonBorderWidth}" BorderColor="{Style HelpButtonBorderColor}"
CornerRadius="{Style HelpButtonCornerRadius}" OnClick="{Action
ToggleVisibility('Help')}">
```

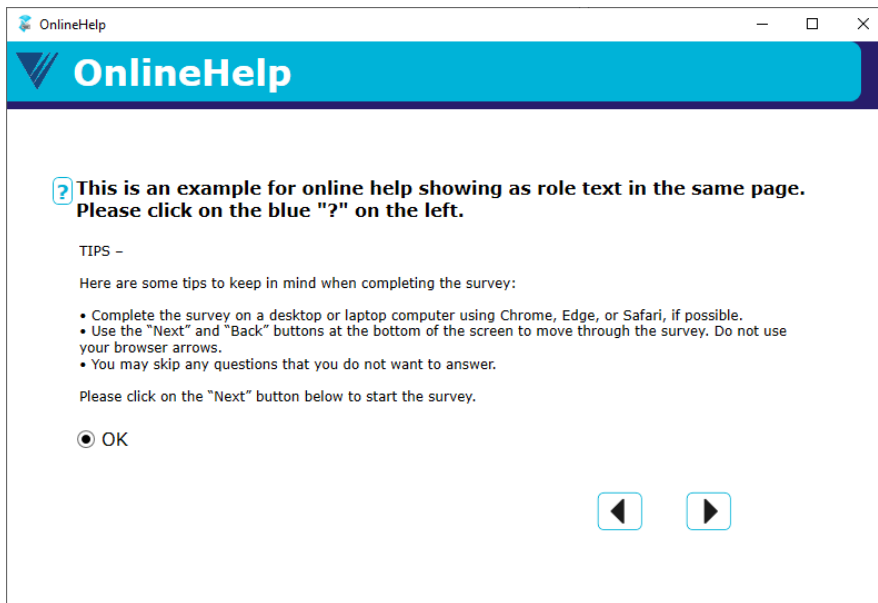

In the data model, we define the role text “HelpText” for the field.

```
Q5 (Q5)
"This is an example for online help showing as role text in the same page. Please click on the blue "?" on the left."
HelpText "<p><strong>TIPS -</strong></p>
<br>Here are some tips to keep in mind when completing the survey:
<br><br>• Complete the survey on a desktop or laptop computer using Chrome, Edge, or Safari, if possible.
<br>• Use the "Next" and "Back" buttons at the bottom of the screen to move through the survey. Do not use your browser arrows.
<br>• You may skip any questions that you do not want to answer.
<br><br>Please click on the "Next" button below to start the survey.<br>"
: (OK)
```

As a result, we have a blue “?” button to the left of the question text.



When the blue “?” button is clicked, help text is displayed below the question text.



This is not limited to the field level. For instance, we can add the same help button and field text in a category template, and then add role text to the type library so that the help text becomes answer category level.

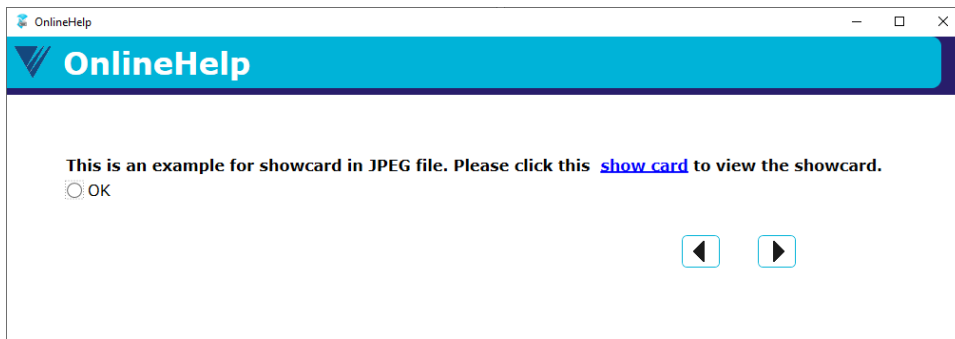
Although this is usually the way we show help text in web surveys, this can be applied to interviewer-administered surveys in the same way.

4.4 Hyperlink in Field Text to Display Showcard

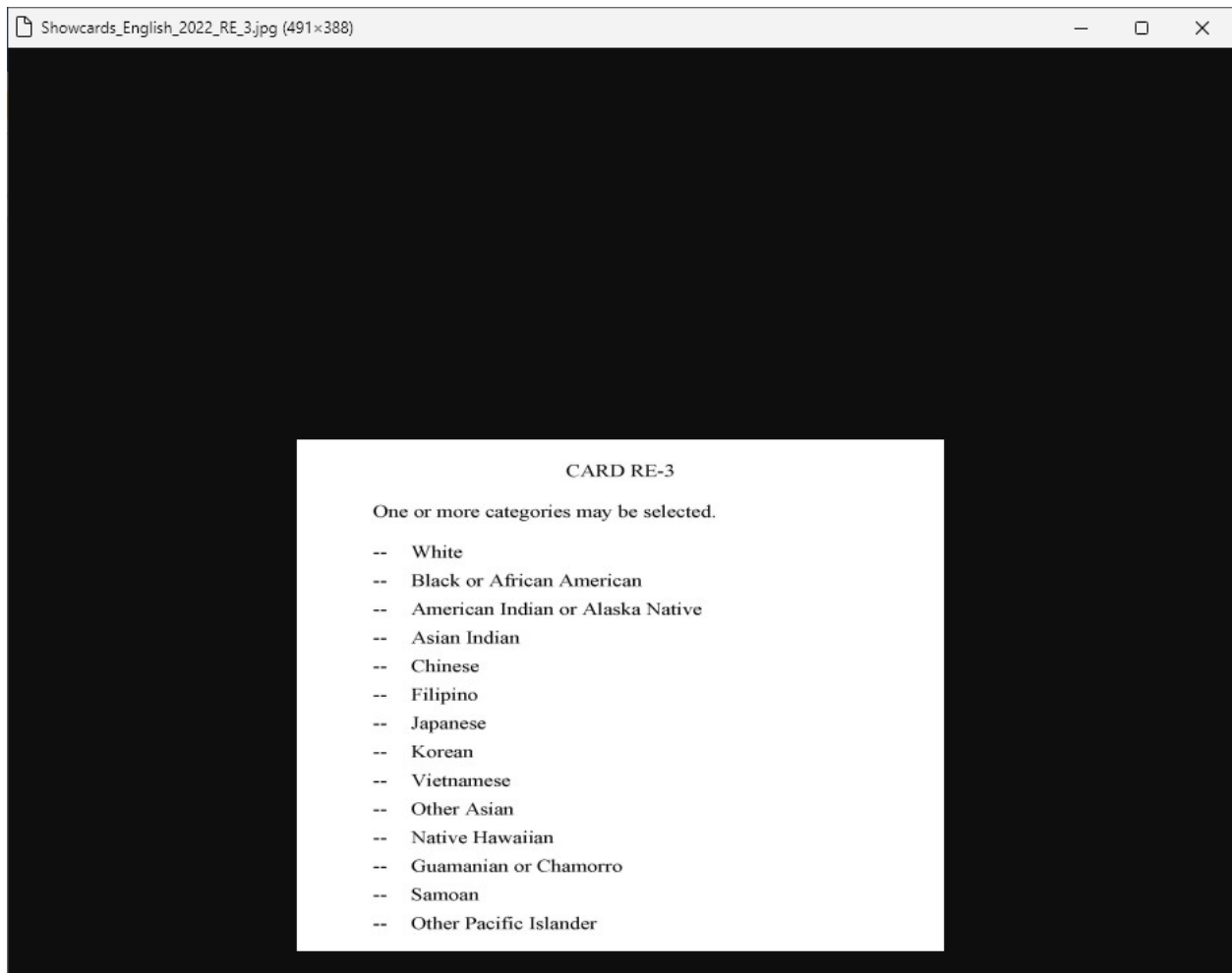
In face-to-face surveys, showcards are often used to present information to respondents. Either we choose to present the showcards on our devices, or simply use that as the means for the interviewers to verify the correct showcard. There is value in having the ability to display the showcards along with the questions on the screen during the survey.

This time, we will do everything in the field text and not bother with the resource database.

```
Q6 (6)
"This is an example for showcard in JPEG file. Please click this <blue>
<hyperlink
  OnClick="{Action
    StartLocalProcess('C:\\Program Files (x86)\\Microsoft\\Edge\\Application\\msedge.exe',
      '--app=C:\\Documents\\Showcards\\Showcards_English_2022_RE_3.jpg',
      '', True, Normal)}\"
  >show card</hyperlink></blue> to view the showcard."
: (OK)
```



Clicking the hyperlink triggers the StartLocalProcess action to run the local program “msedge.exe” to open a JPEG file in MS Edge.



5. Conclusions

We have seen many different ways to display supplementary materials.

If the needs are limited (e.g., only help text for a couple questions), we can easily program that in the field text without extra settings in the resource database.

However, for large surveys that need to show a lot of help text or showcards, maintaining all the external texts and images outside of Blaise would be more manageable. In that case, setting up text roles, events, and actions in the resource database up front would be a more efficient way to deal with it. For instance, we only need to use role text to determine the file name or page number in the field definition, and the resource database can take care of the rest.

It is worth noting that, as helpful as it is, the StartLocalProcess action works only in Windows, not in browsers.

On the other hand, in web surveys, the rule of thumb would still be avoiding popups, whether we can start a local process or not. In this case, using field text to show help text is more desirable. It should be noted that survey designers might need to beware of too much help text displayed on the page that becomes more of a distraction than a help.