Blaise

Tips & Tricks

Overview

- Constants
  - Auxfield vs. Constant Examples
- Snippets
- Debugging the Rules
- Debugging Layout
  - Finding a field
  - Using Visual Cues
  - Browser Preview
  - Warning Icon
  - Validate
  - Sizing Issues
  - Layout Instructions

Constants

Constants

- Similar to constants in other programming languages
  - Cannot be reassigned, only referenced
- Define INTEGER, REAL, ARRAY, BOOLEAN, STRING
  - Choose to define whether a string is translatable
    - Double quotes: can define multiple languages in .blax/.bitt
    - Single quotes: cannot be translated (useful for enhanced text)
- Use constants in other type declarations
- Cleaner way of initializing variables
- Can also be used in Manipula

Constants

```
CONST
   Insanity = 'Repeating the same text.'
   Veto = FALSE
   SurveyDay = [ 'Monday', 'Wednesday', 'Friday' ]
   SomeDays = [ ENG "Monday" "Maandag", "Tuesday" "Dinsdag", ENG "Wednesday" NED "Woensdag" ]
   Root2 = 1.41421356
   OK = TRUE
   Zero = 0
   Spacing = '<NewLine count=3>'
   ASize = 10
   Code = 3
   Min = 1.25
   Max = 9.75
   txtHe = ENG "he" FRA "il"
```

Auxfield vs. Constant Examples

Constants

```
HowManyPeople
IF HowManyPeople = 1 THEN
   IF ACTIVELANGUAGE = EN THEN
      IntroText1 := '<newline>Hello ' + Name+ ' and welcome to the questionnaire.'
      IntroText2 := '<newline count=3>I would like to now ask you some questions about your living situation.'
   ELSE // ACTIVELANGUAGE = NL
      IntroText1 := '<newline>Hallo '+ Name+ ' en welkom bij de vragenlijst.'
      IntroText2 := '<newline count=3>Ik wil je nu vragen stellen over jouw woonsituatie.'
   ENDIF
ELSE
   IF ACTIVELANGUAGE = EN THEN
      IntroText1 := '<newline>Hello, welcome to the questionnaire.'
      IntroText2 := '<newline count=3>I would like to now ask each of you some questions about your living situations.'
   ELSE // ACTIVELANGUAGE = NL
      IntroText1 := '<newline>Hallo, welkom bij de vragenlijst.'
      IntroText2 := '<newline count=3>Ik wil jullie nu vragen stellen over jullie woonsituatie.'
   ENDIF
ENDIF
```

Constants

```
CONST
   // These are translatable through a BITT file because of the double quotes
   cIntroText1Single = EN "<newline>Hello $name and welcome to the questionnaire."
                       NL "<newline>Hallo $name en welkom bij de vragenlijst."
   cIntroText2Single = EN "<newline count=3>I would like to now ask you some questions about your living situation."
                       NL "<newline count=3>Ik wil je nu vragen stellen over jouw woonsituatie."
   cIntroText1Multi =  EN "<newline>Hello, welcome to the questionnaire."
                       NL "<newline>Hallo, welkom bij de vragenlijst."
   cIntroText2Multi =  EN "<newline count=3>I would like to now ask each of you some questions about your living situations."
                       NL "<newline count=3>Ik wil jullie nu vragen stellen over jullie woonsituatie."

...


HowManyPeople
   IF HowManyPeople = 1 THEN
      IntroText1 := REPLACE(cIntroText1Single, '$name', Name)
      IntroText2 := cIntroText2Single
   ELSE
      IntroText1 := cIntroText1Multi
      IntroText2 := cIntroText2Multi
   ENDIF
```

Constants

```
AUXFIELDS
   HelpUrl : STRING
RULES
   HelpUrl := '<hyperlink uri=http://help.blaise.com target=_blank>Blaise Help</hyperlink>'
```

```
CONST
   // This is not translatable through a BITT because of the single quotes
   cHelpUrl = '<hyperlink uri=http://help.blaise.com target=_blank>Blaise Help</hyperlink>'
```

Constants

```
TYPE
   TRange = 1..10

LOCALS
   i : Trange

AUXFIELDS
   Max : INTEGER

FIELDS
   People : ARRAY[1..10] OF bPerson

RULES
   Max := 10
   HowMany
   IF HowMany = Max THEN
      MoreThanMax
      IF MoreThanMax = YES THEN
         MoreThanMaxInfoText
      ENDIF
   ENDIF
   FOR i := 1 TO HowMany DO
      People[i](i)
   ENDDO
```

```
CONST cMax = 10

TYPE
   TRange = 1..cMax

LOCALS
   i : TRange

FIELDS
   People : ARRAY[1..cMax] OF bPerson

RULES
   HowMany
   IF HowMany = cMax THEN
      MoreThanMax
      IF MoreThanMax = YES THEN
         MoreThanMaxInfoText
      ENDIF
   ENDIF
   FOR i := 1 TO HowMany DO
      People[i](i)
   ENDDO
```

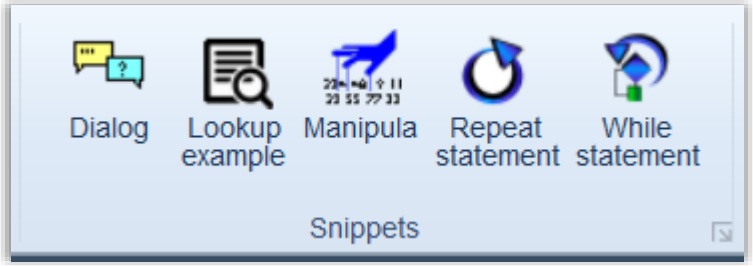
Snippets

Snippets

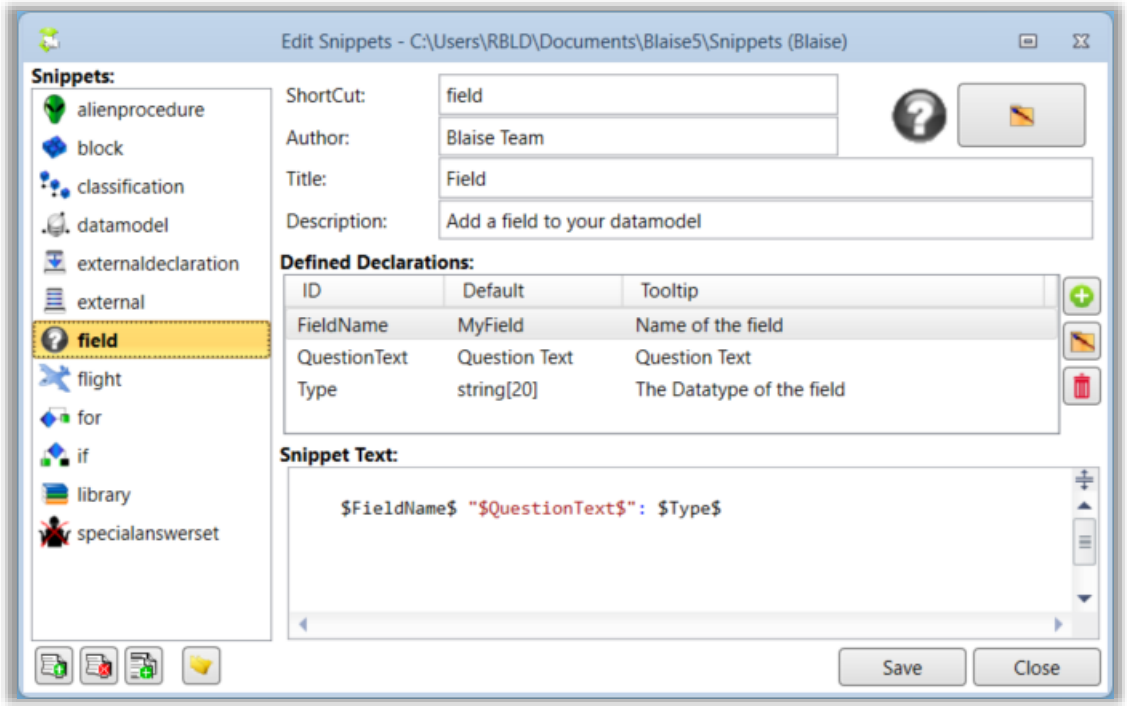
- Small, reusable pieces of code that can quickly be inserted
  - Separate snippets for Manipula and Data model projects
- Can be inserted through button click or a shortcut
- Useful if you need code directly in the sources instead of an Include file
- Can be shared throughout organization
- Improve programming speed

Snippets

Alien procedure
Block
Classification
Datamodel
External declaration
External in rules
Field
Flight Survey
For loop
If statement
Library
Special Answer Set
Snippets


Dialog
Lookup example
Manipula
Repeat statement
While statement
Snippets

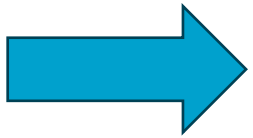
Snippets



- ShortCut: use shortcut + TAB to insert the snippet
- Title: Text in the Control Cenre Ribbon
  - Can optionally add a thumbnail
- Declarations
  - Parts of snippet that can be changed by user
  - Can be reused throughout snippet
  - Navigated via TAB

Snippets

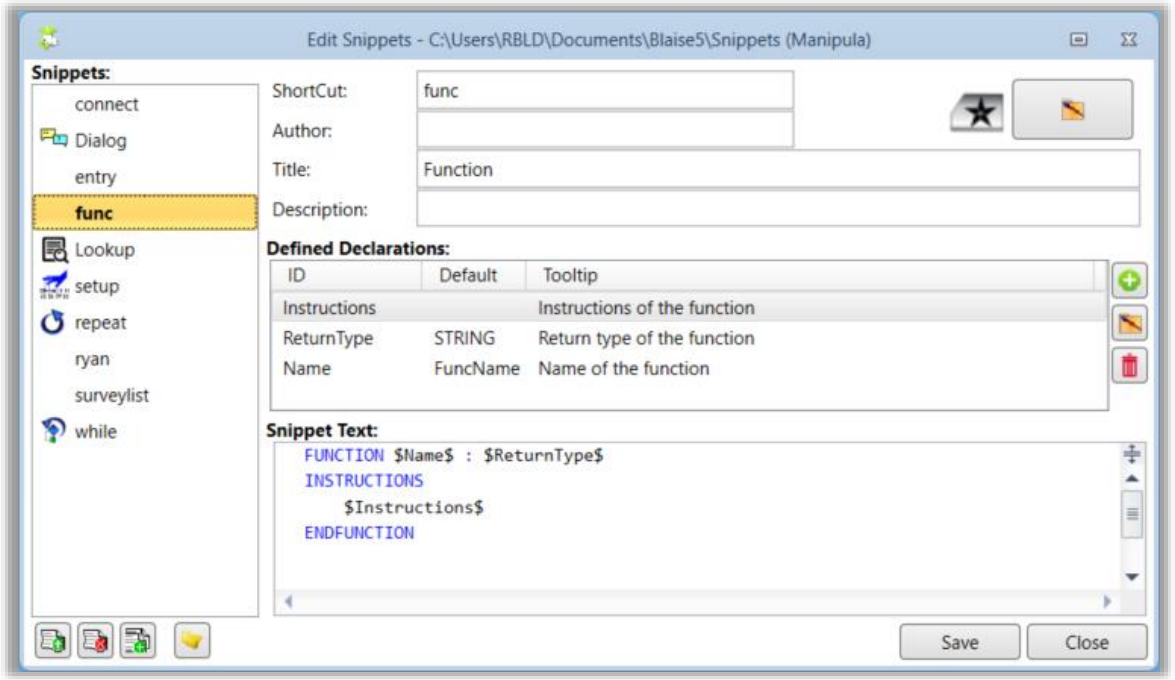
```
DATAMODEL SnippetDatamodel

PRIMARY ID

FIELDS

  ID : INTEGER[4]
field

ENDMODEL
```

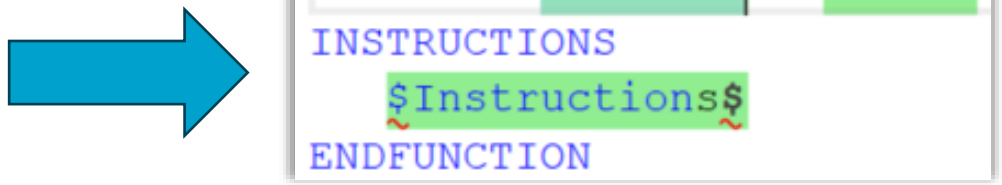


```
DATAMODEL SnippetDatamodel

PRIMARY ID

FIELDS

  ID : INTEGER[4]

  MyField "Question Text": string[20]


ENDMODEL
```

Snippets

- Create your own!





- Tip: Use Backspace + ESC to exit snippet editing in the middle of the function

Debugging the Rules

Debugging the Rules

Session Viewer

- View current status of a session
- Refresh to view changes since last update
- View the values of Auxfields
- Opened via Control Centre, Server Manager or executable

Requirements

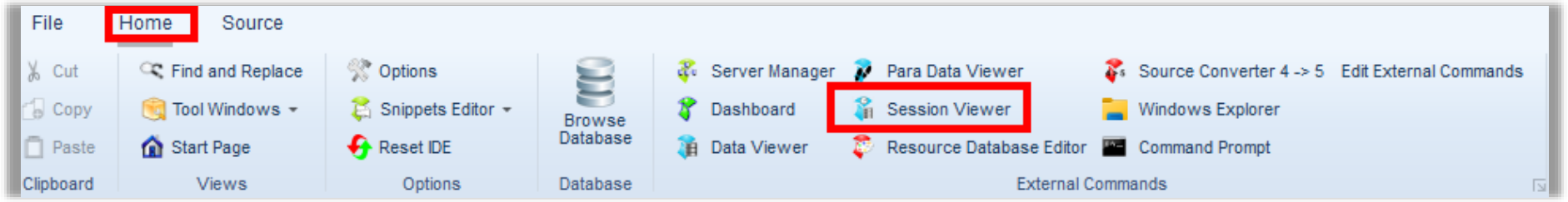
- Data model must have a primary key

Tips & Tricks

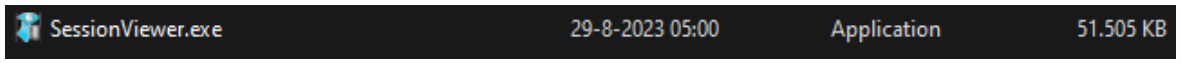
- Data entry settings
  - Set Server Contact to on field switch
- Critical fields

Debugging the Rules

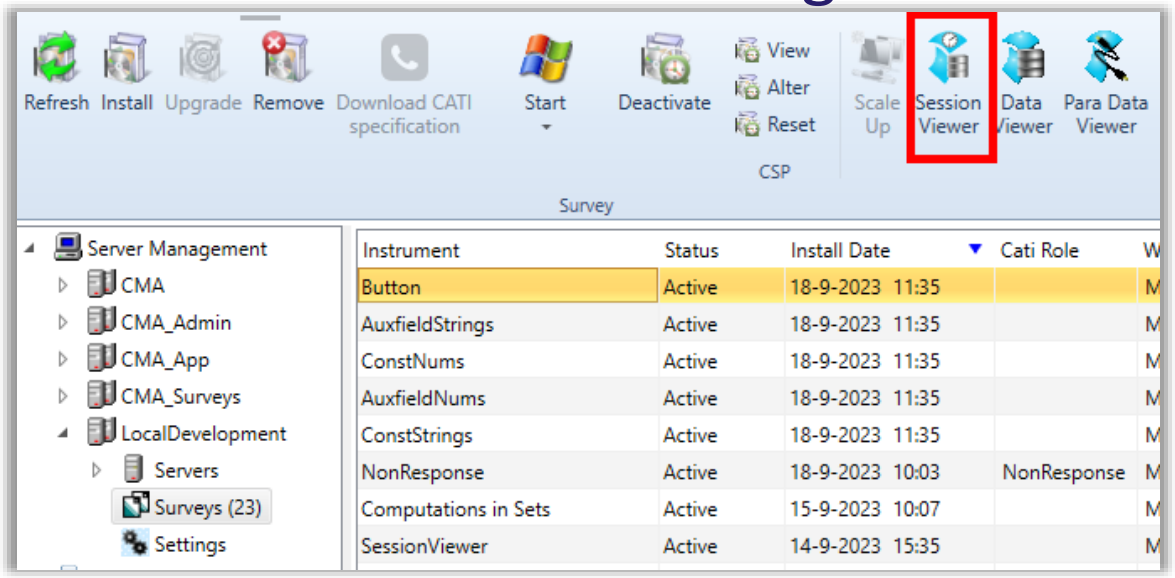
Control Centre



SessionViewer.exe



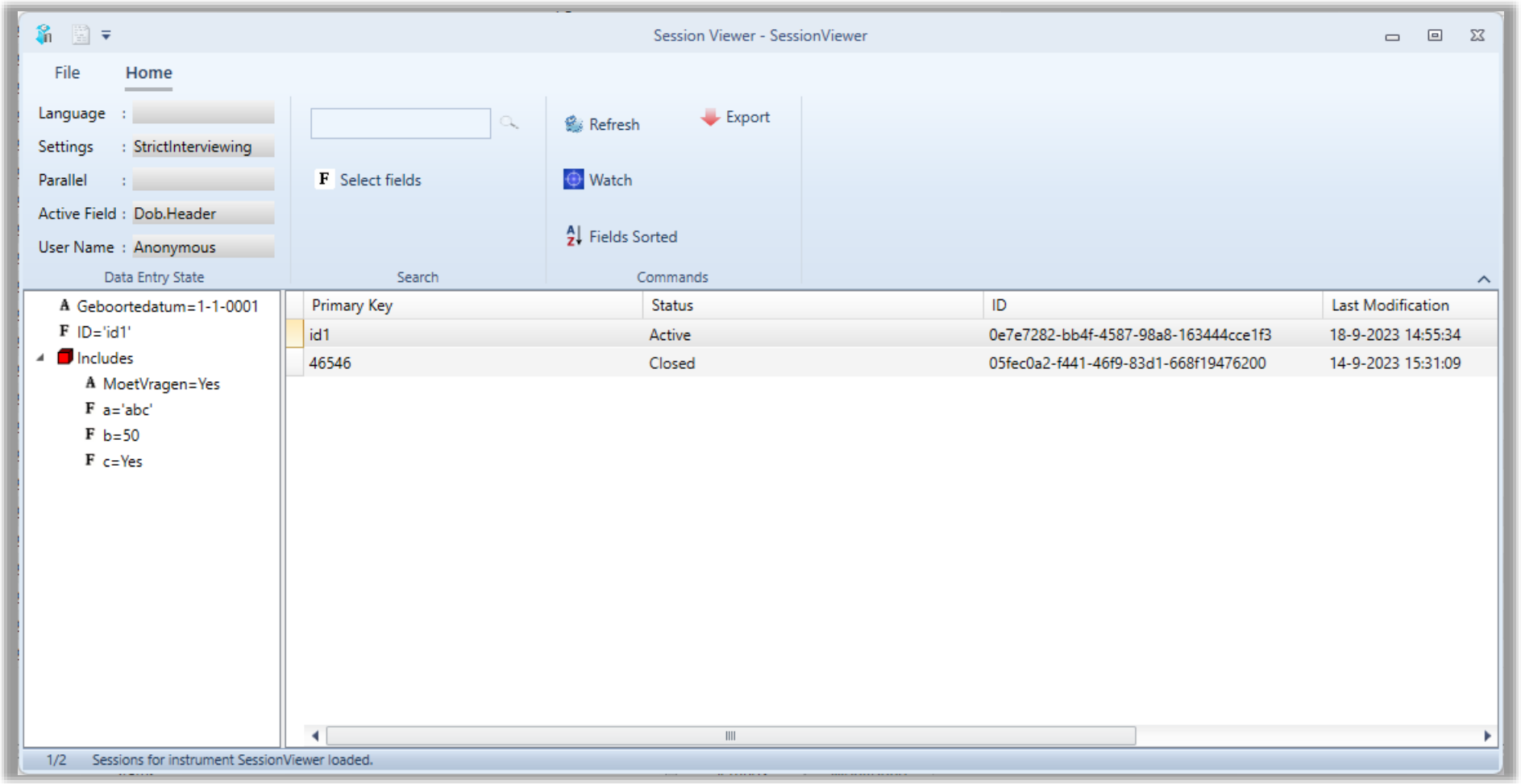
Server Manager



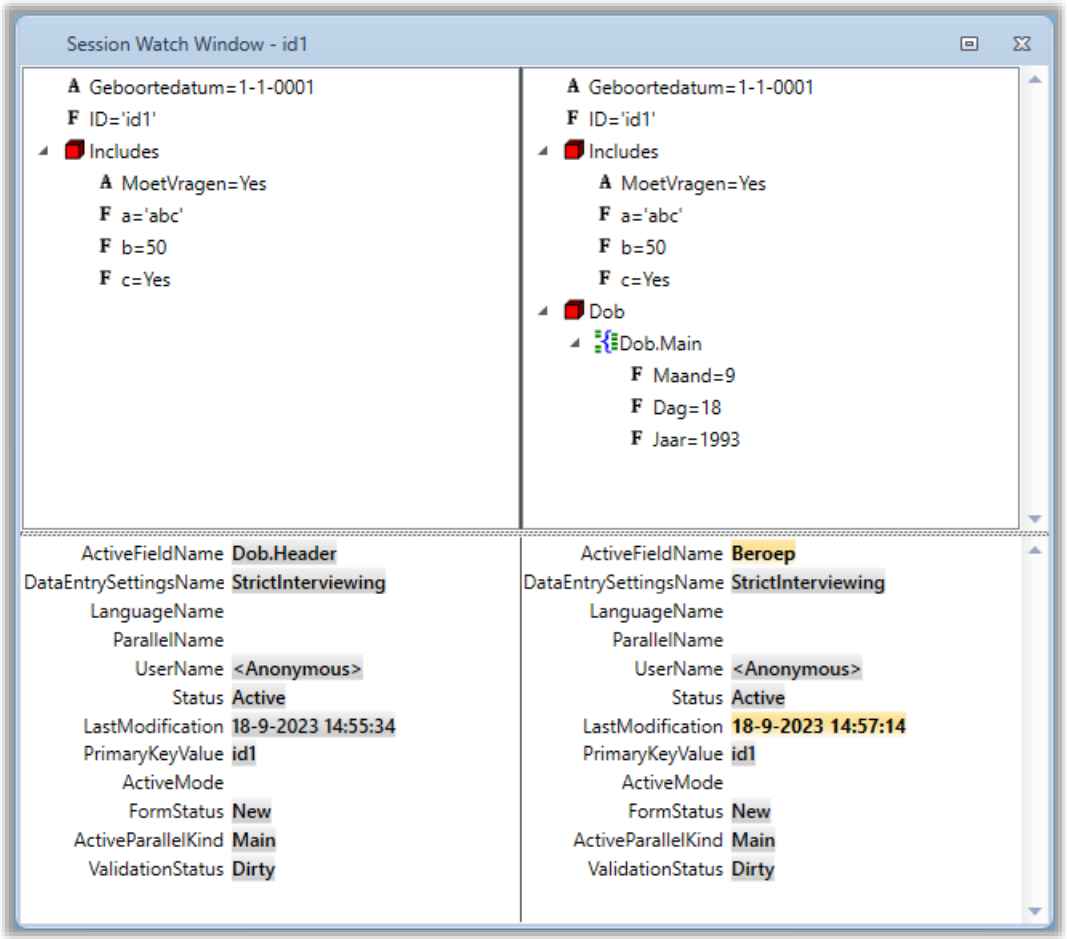
Debugging the Rules

```
RULES
   Includes.KEEP
   Dob
   GeboorteDatum := TODATE(Dob.Dag, Dob.Maand, Dob.Jaar)
   gBeroep
   IF Beroep in [1, 3] {ASK Field} AND Includes.c = Yes {Included field} THEN
      IF GeboorteDatum.Age {Calculated Auxfield} < Includes.b {Included Field} AND Includes.a = RESPONSE {Included Field} THEN
         IF Includes.MoetVragen = YES {Included Auxfield} THEN
            ProgramLang
         ENDIF
      ENDIF
   ENDIF
```

Debugging the Rules



Debugging the Rules



Debugging the Rules

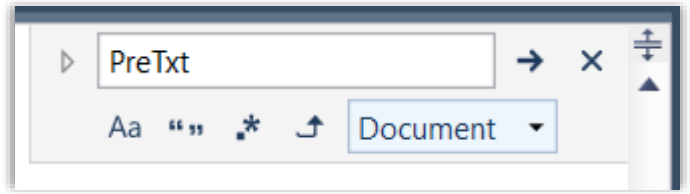
```
RULES
   Includes.KEEP
   Dob
   GeboorteDatum := TODATE(Dob.Dag, Dob.Maand, Dob.Jaar)
   gBeroep
   IF Beroep in [1, 3] {ASK Field} AND Includes.c = Yes {Included field} THEN
      IF GeboorteDatum.Age {Calculated Auxfield} < Includes.b {Included Field} AND Includes.a = RESPONSE {Included Field} THEN
         IF Includes.MoetVragen = YES {Included Auxfield} THEN
            ProgramLang
         ENDIF
      ENDIF
   ENDIF
```

Debugging Layout

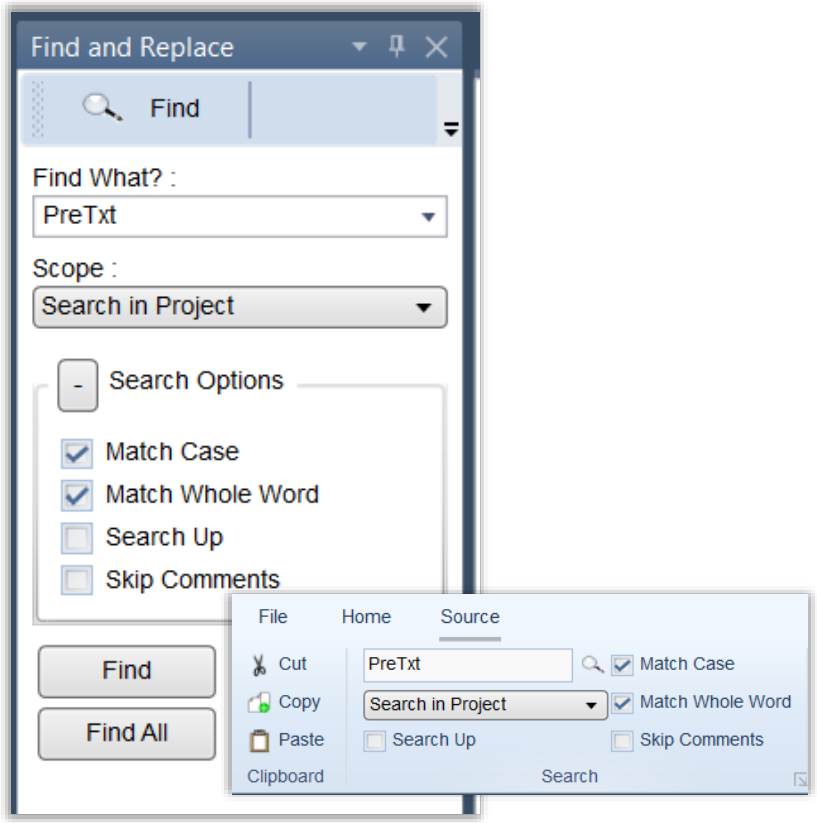
Finding a field

Find a field

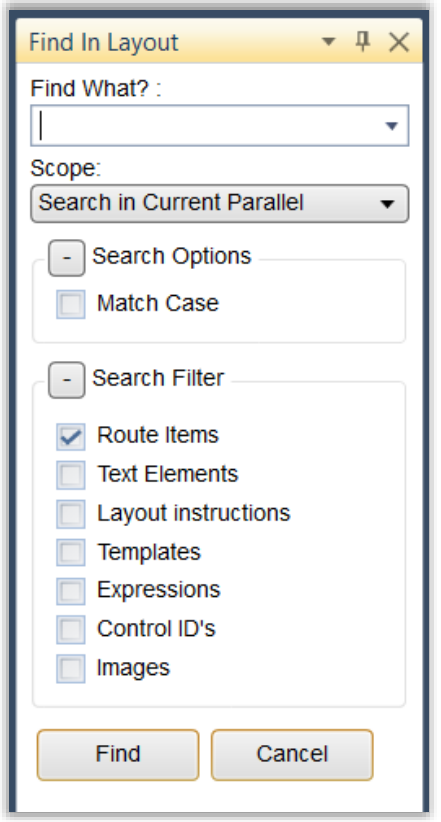
CTRL + F (Source View)



CTRL + SHIFT + F (Source View)

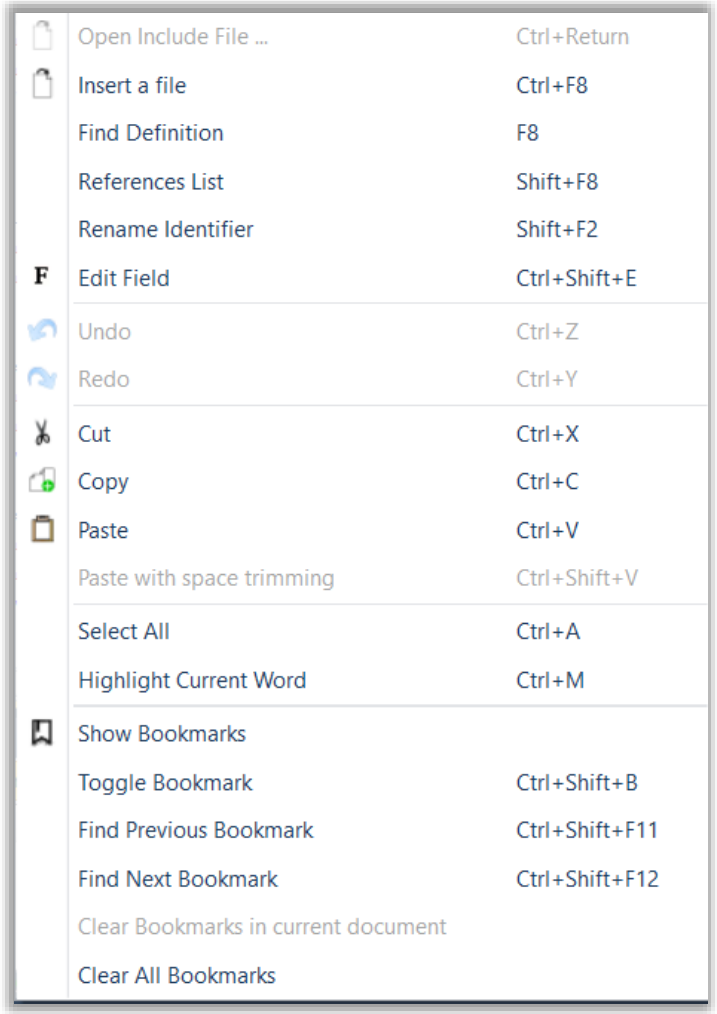


CTRL + F (Layout Designer)



Find a field definition

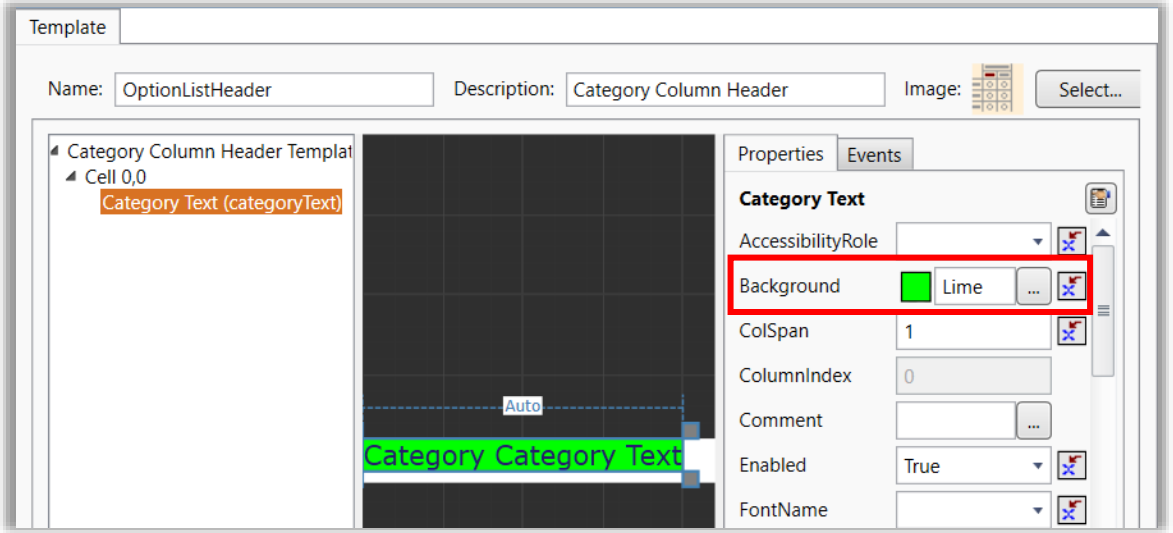
- CTRL + F7
  - Quickly jump from the focused field in the Layout Designer to the same field in the Source View or back
    - It jumps to the location where the field is asked or shown (in the rules)
- F8
  - Jumps to the definition of a field
- Right-click on an identifier to see all the options in its context menu

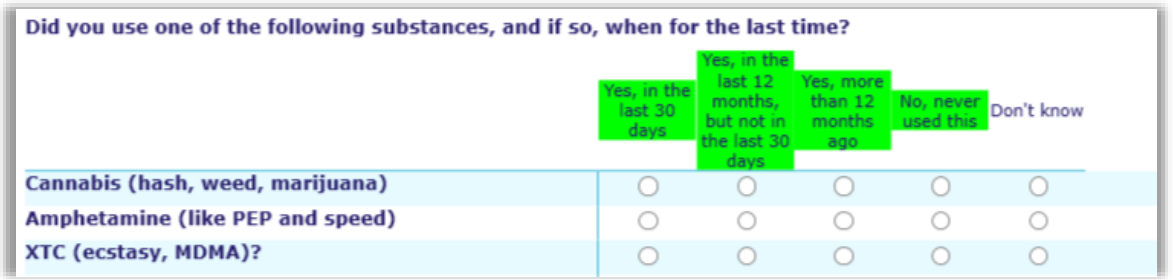


Using visual cues

Visual Cues

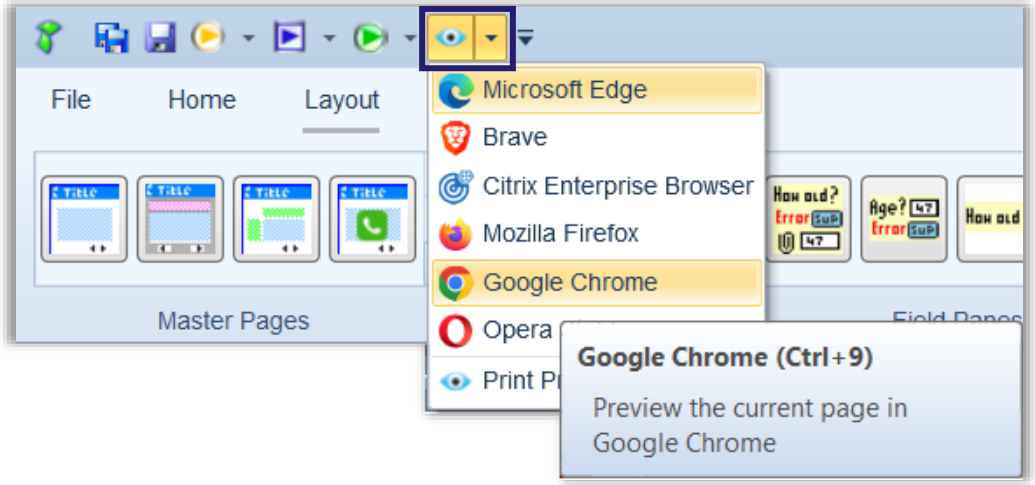
- Change a template visually
  - Add different colors to parts of templates
  - Change sizes
  - Add an extra element
- This will allow you to see if you're changing things in the template you were expecting
  - Are you maybe changing it in the wrong Resource Set?
- It will also show where each part of a template is presented, in relation to each other
  - This can help when trying to determine which cell needs a different size





Browser Preview

Browser Preview

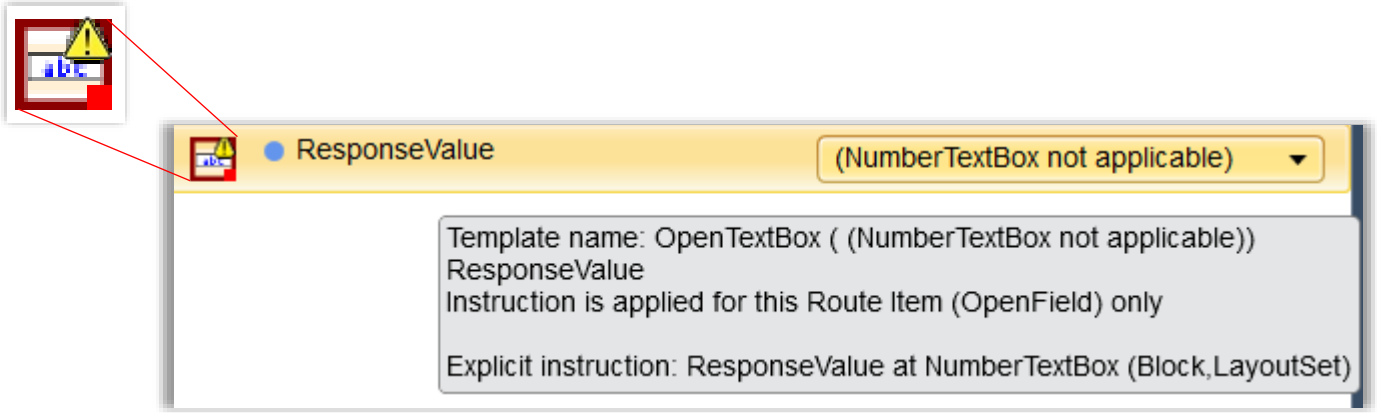


- The Layout Designer shows a preview of the DEP
- To see a preview in the browser, click the dropdown next to the eye in the quick menu bar and select the browser
  - The current page will open in your selected browser
  - Only visual, NOT behavior
  - You can open the page in multiple browsers at the same time
    - check if everything looks alright in all browsers
  - Refresh the browser(s) (F5) to update the page after changing something in the Layout Designer to see the result of that change

Warning Icon

Warning Icon

- Shows that a template is not applicable
  - Because the type is wrong
  - Or the template does not exist in the Resource Database
- Hovering over the icon will show a tooltip with details



Validate

Validating your data model

- Validation is important to make sure everything works as designed
  - Invalid sources won't compile/run
  - Invalid layout may cause unexpected behavior
- Validating consists of three steps:
  - Step 1: Validate your sources
  - Step 2: Validate your Resource Database
  - Step 3: Validate the layout
- If at any point something does not validate
  - Fix it
    - Start validating again from the level in which you fixed it (sources/Resource Database/Layout Designer)

Validate - Sources

- The first step is validating your sources
  - By default, this is done automatically by the parser
    - If you turned this option off, you need to compile the project manually (yellow arrow)
  - If you are unsure if something was updated after a change
    - Type a space somewhere
    - Compile manually (yellow arrow)
      - This will force the system to recreate your .bmix
      - Last resort: restart the services (if possible)

- The error window will show errors, warnings and information
  - Errors: need to be solved
  - Warnings: read them, consider if they will affect your questionnaire
    - Fix if they may affect your input/output values
    - It’s ok to leave them if they do not apply to what you are using it for
  - Information: read and decide if you need to do something with it or not

Validate – Resource Database

- When your sources compile correctly, check your layout
  - Validate your Resource Database
    - Fix all errors
      - Preferably all, but the ones pertaining to templates you use at the very least
    - Fix all warnings
      - Preferably all, but the ones pertaining to templates you use at the very least
    - Why all?
      - Because you may decide to use one of the invalid templates at a later point or it may be automatically applied if you change your sources and invalid templates may cause unexpected behavior

Validate – Layout Designer

- If your sources compile and your Resource Database is valid, your layout may still show errors
  - Because the combination of templates you ended up using causes errors
    - Often these are Auto-Stretch errors
    - Sometimes these are Scrollbar-warnings

Validate – Layout Designer

- To fix:
  - click on the error message
    - It will jump to the template where it encountered the unwanted situation
    - Note that the error usually requires changing something in this template as well as in one or more parent templates (all the way up to the MasterPage template)
      - Use the Layout Designer | Layout instructions or the Template Explorer to see which templates are used
      - Jump to the template in the Resource Database from the Layout Designer
        - This will ensure you are changing things in the correct template
        - If you only need to fix something for one or 2 questions, it may be better to add a parameter to the template and change the value of that
          - This will ensure your other questions don't use that 'fix'
  - If you made changes in the Resource Database, revalidate that
  - If you made the changes in the Layout Designer only, revalidate the layout there

Sizing Issues

Sizing

"Do not stretch in an Auto"

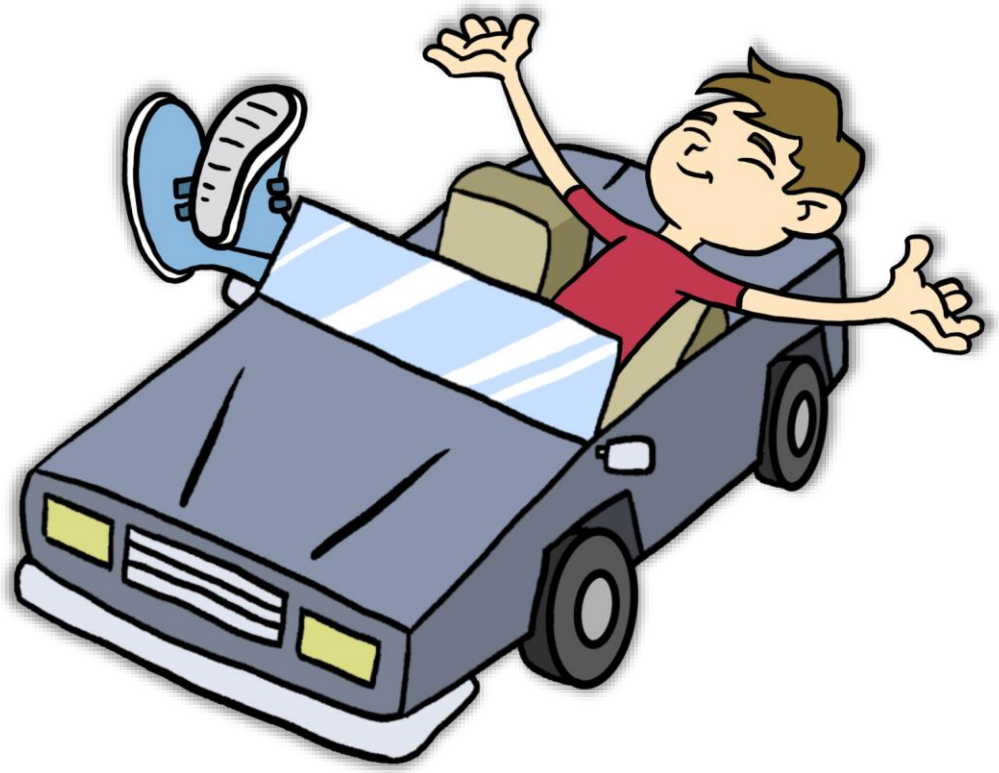


Image credit: Roel Leclair, 2023

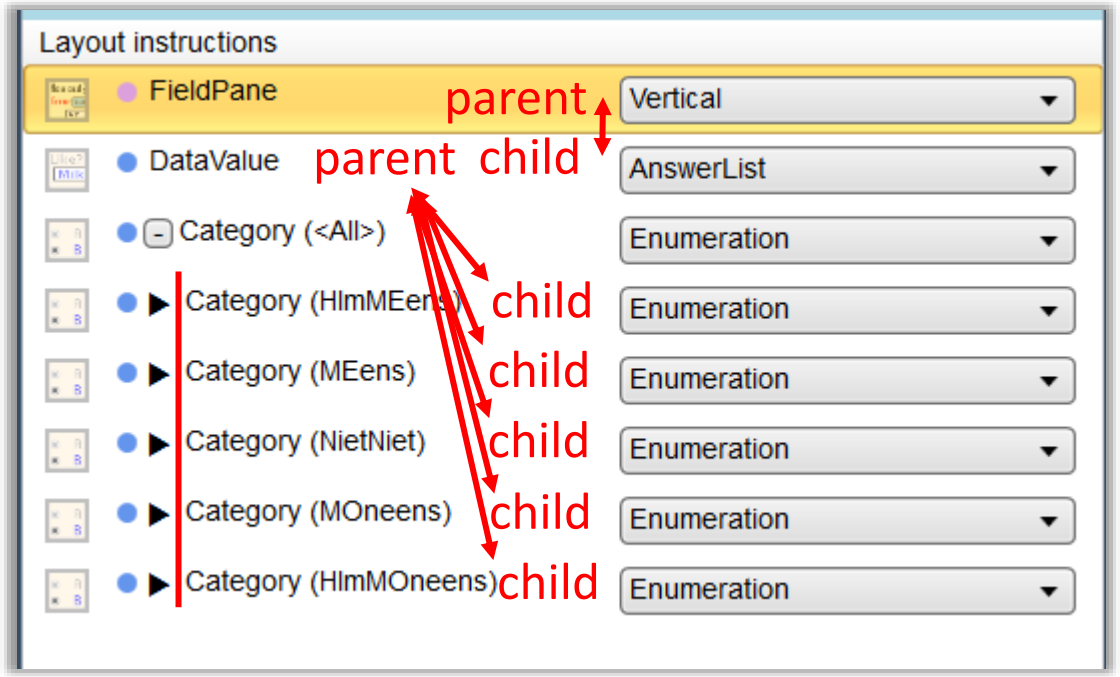
Sizing-table

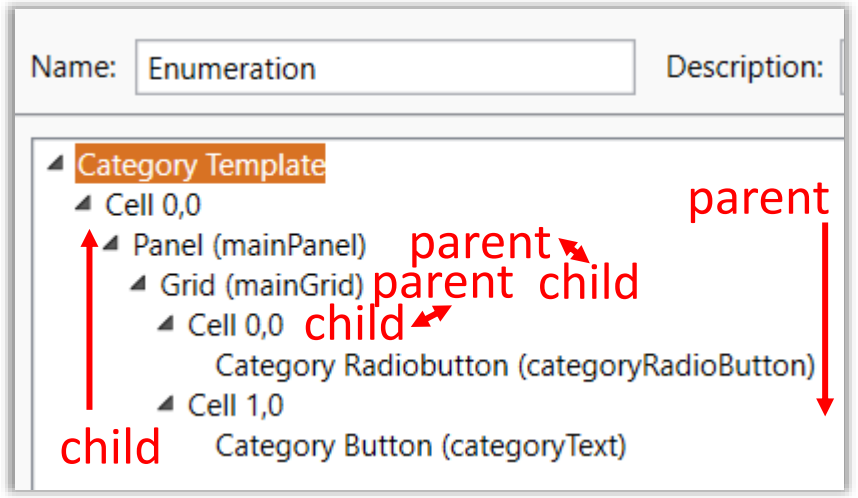
- <child> in <parent> = <cell-value>

		PARENT				
		Auto	**Stretch/***	**%**	**Undefined**	**Fixed size**
	Auto	OK	OK, but if the child is larger than the available space, it will be cut off	OK, but if the child is larger than the available space, it will be cut off	OK	OK, but if the child is larger than the fixed size, it will be cut off
	Stretch/*	NOT OK	OK	NOT OK	OK, if the child directly replaces a placeholder. If not: NOT OK	OK
CHILD	**%**	NOT OK	NOT OK	OK	OK, if the child directly replaces a placeholder. If not: NOT OK	OK
	Undefined	OK if its direct child is Auto or fixed. NOT OK if its direct child is Stretch/*	OK, but if its direct child is Auto or fixed, and larger than the available space, it will be cut off	OK, but if its direct child is Auto or fixed, and larger than the available space, it will be cut off NOT OK if its direct child is Stretch/*	OK, but if its direct child is Auto or fixed, and larger than the available space, it will be cut off NOT OK if its direct child is Stretch/*	OK, but if its direct child is Auto or fixed, and larger than the available space, it will be cut off
	Fixed Size	OK	OK, but may be cut off if there is not enough available space	OK, but may be cut off if there is not enough available space	OK, but may be cut off if there is not enough available space	OK, but may be cut off if there is not enough available space

Child-Parent relationship examples

The DataValue template is child to the FieldPane template,
but also parent to all the Category templates

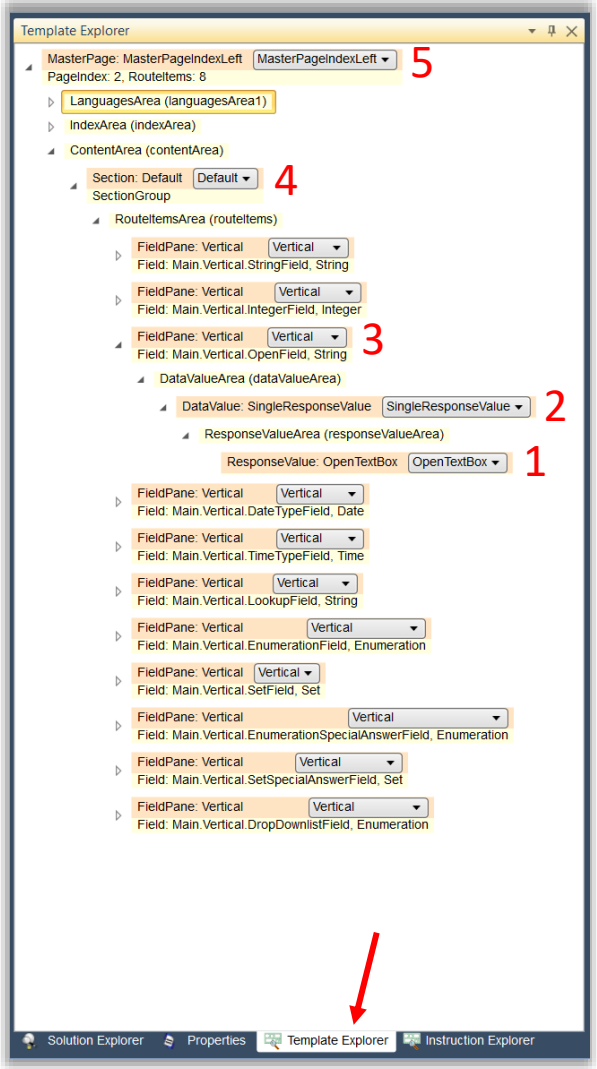




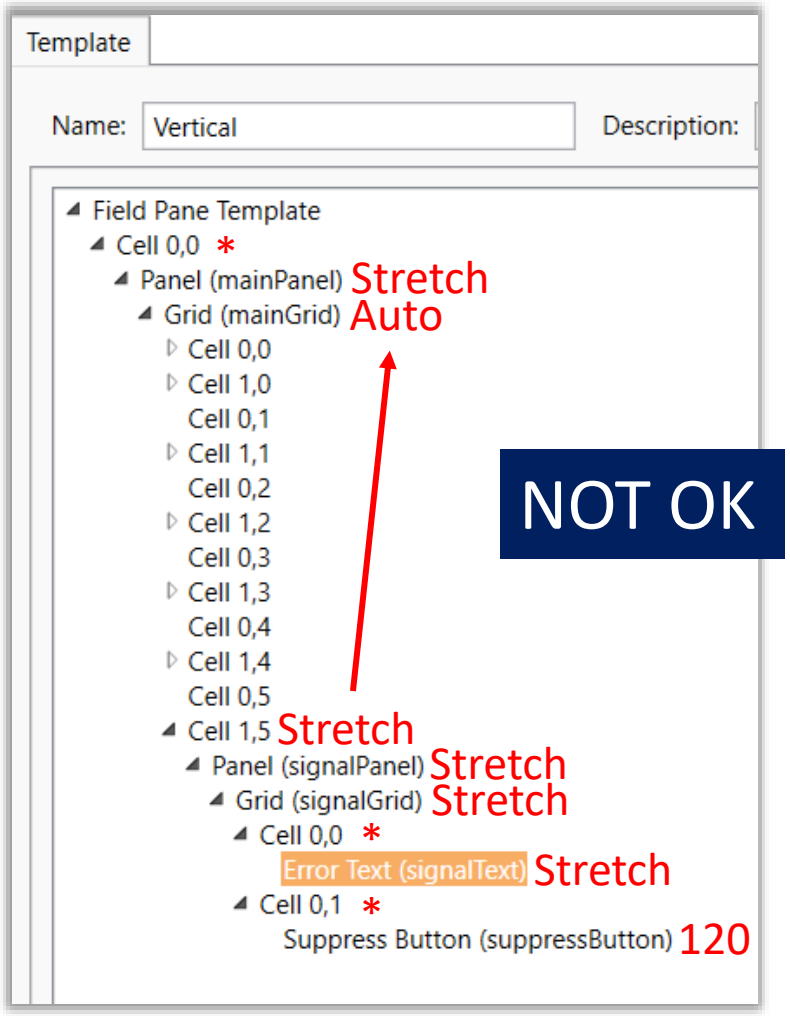
mainGrid is parent to Cell 0,0
but also child to mainPanel

Sizing – simplified rules

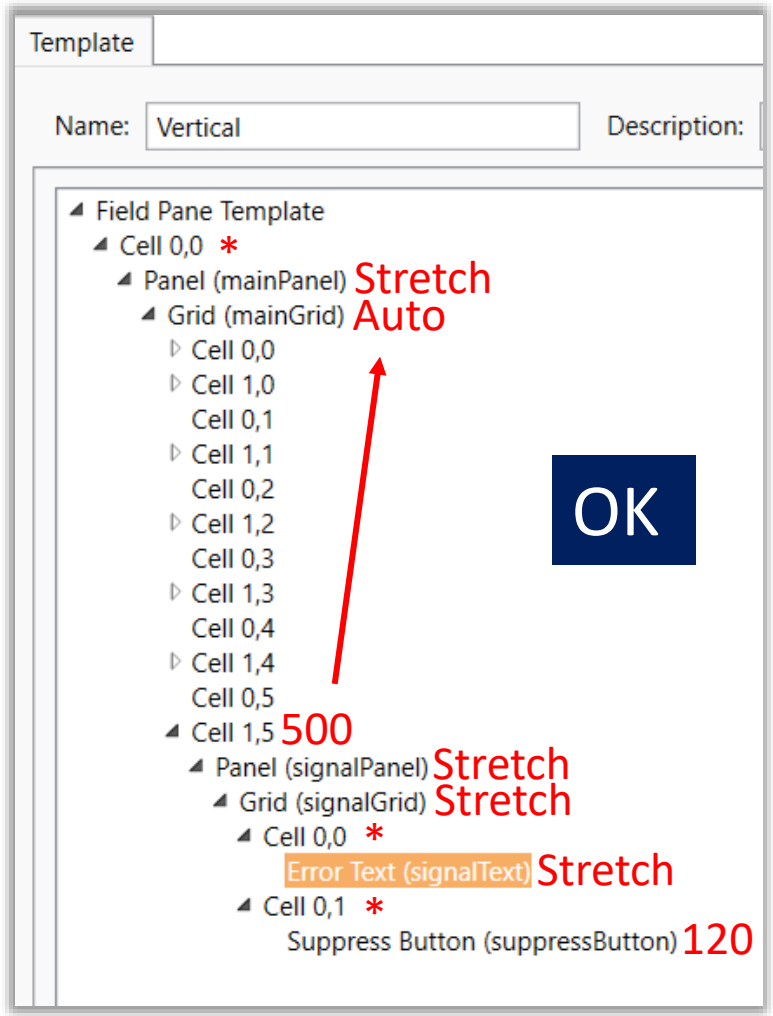
- Start from the bottom and work your way up through the template (and its parent templates)
  - Check the *Template Explorer* to see which templates are used for this field
  - If you want a text to wrap, it needs to be Stretch
    - This means that all its parents should be either Stretch, Star (*) or Fixed
- Note that once a fixed size is introduced, this 'breaks' a chain of Stretch/*
  - its parent can be seen as the new bottom element to start from



Sizing - example



Widths



Sizing – simplified rules

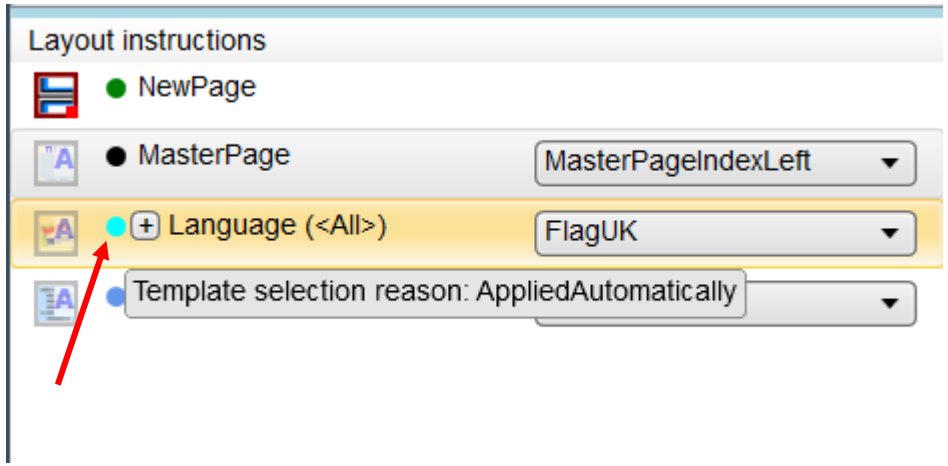
- (Column)Widths are usually set to Stretch
  - so they can wrap where needed and use all available space
  - except maybe on the top level, depending on whether or not you always want your page to be the same width (use a fixed size in that case)
    - Make sure to have a background in case the screen is bigger
- (Row)Heights are usually set to Auto
  - except maybe on the top level, depending on whether or not you want your content area to 'float' or be stationary (use Star (*) in that case)
- Use fixed sizes for elements that need to stay the same size, regardless of the rest of the page
  - E.g. a header, a footer, or (navigation) buttons
- Use margins to space out things instead of an empty row or column
  - This will cut down on the number of elements and can improve performance as a result

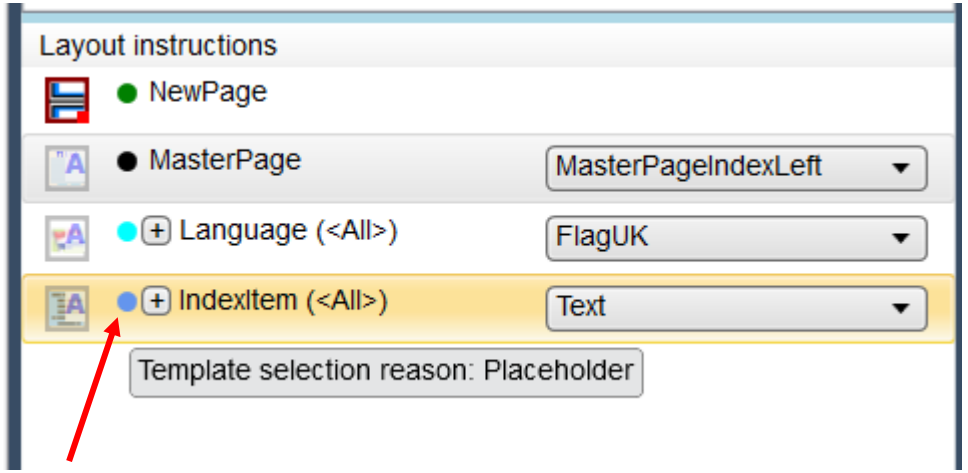
Layout Instructions

Why and where are they used

Layout Instructions – selection reason

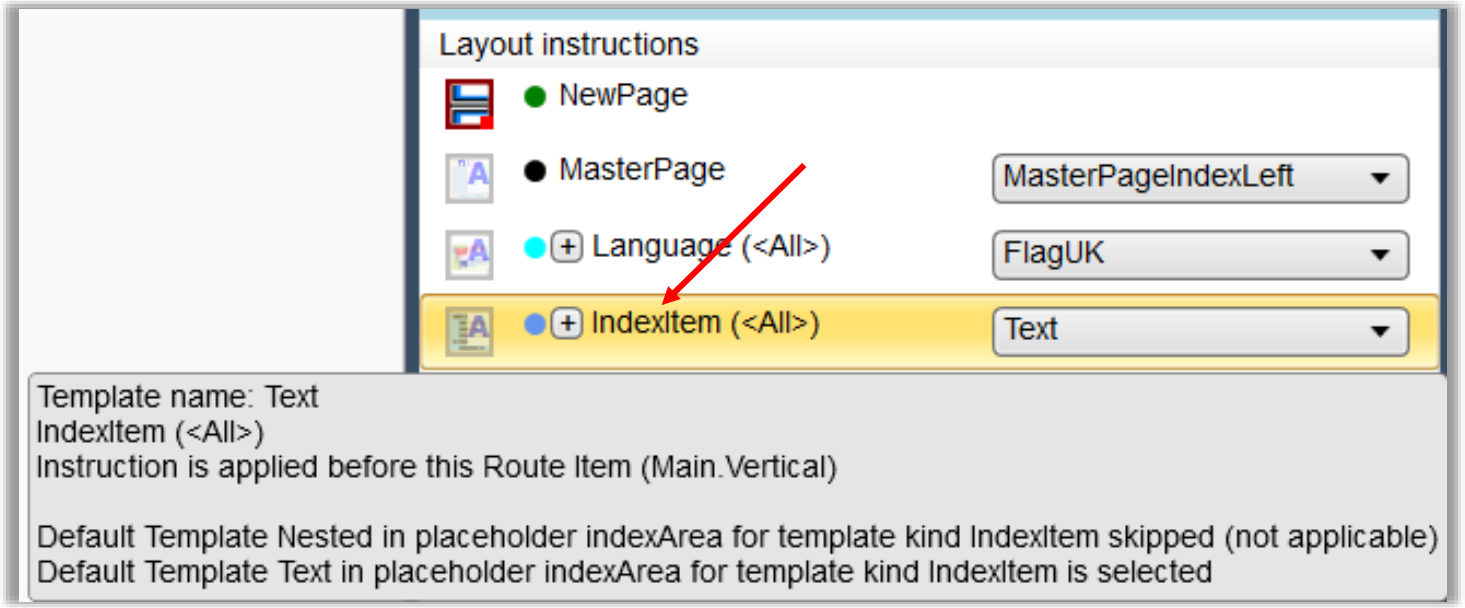
- You can see why a template has been applied
  - By hovering over the colored dot in front of the template in the Layout instructions
  - Check this if you don't understand why a template has been applied





Layout Instructions – selection reason

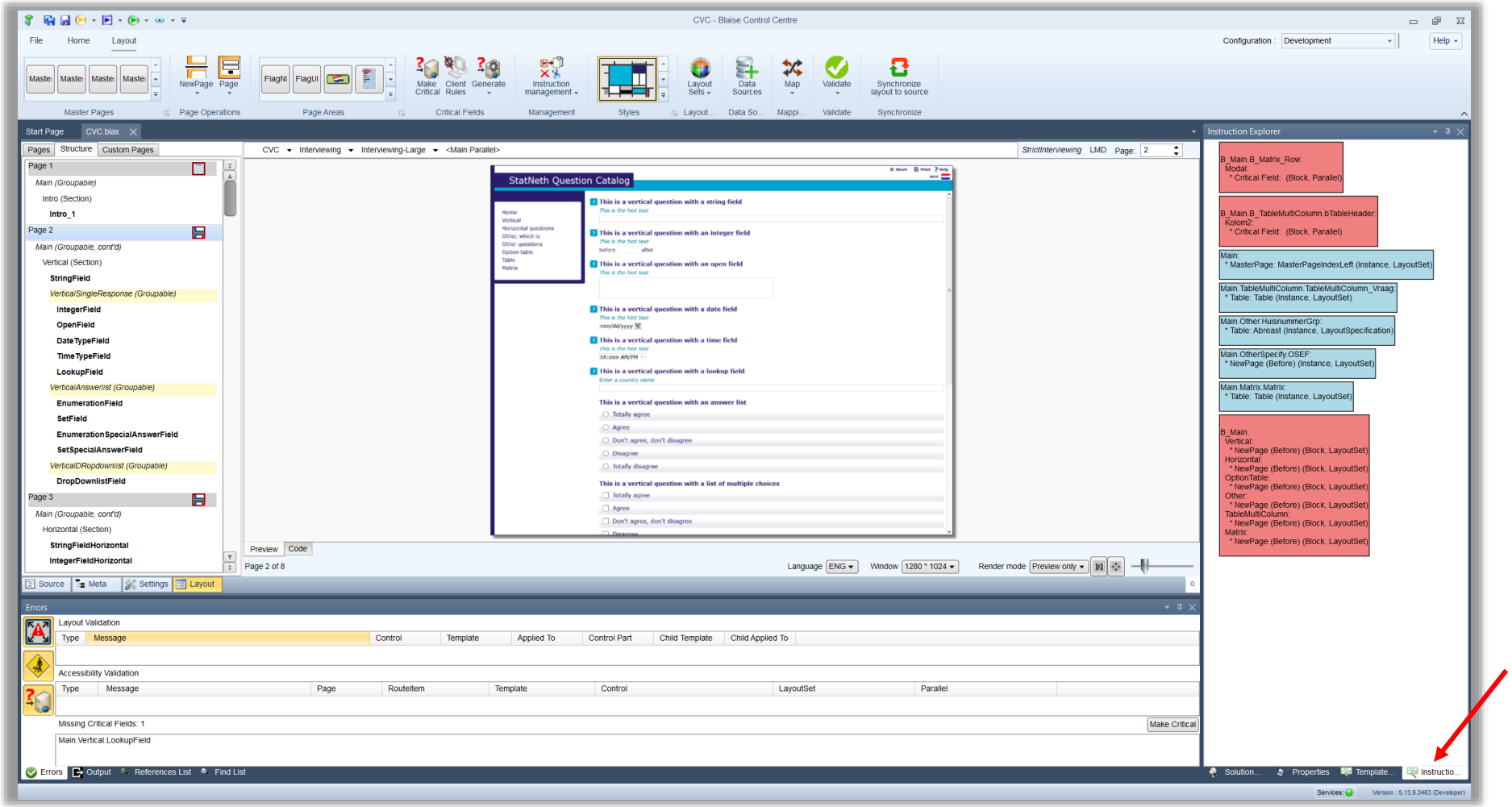
- If you need more information, you can hover the template kind
  - This will show more detailed information
    - It also shows why another template was NOT applied

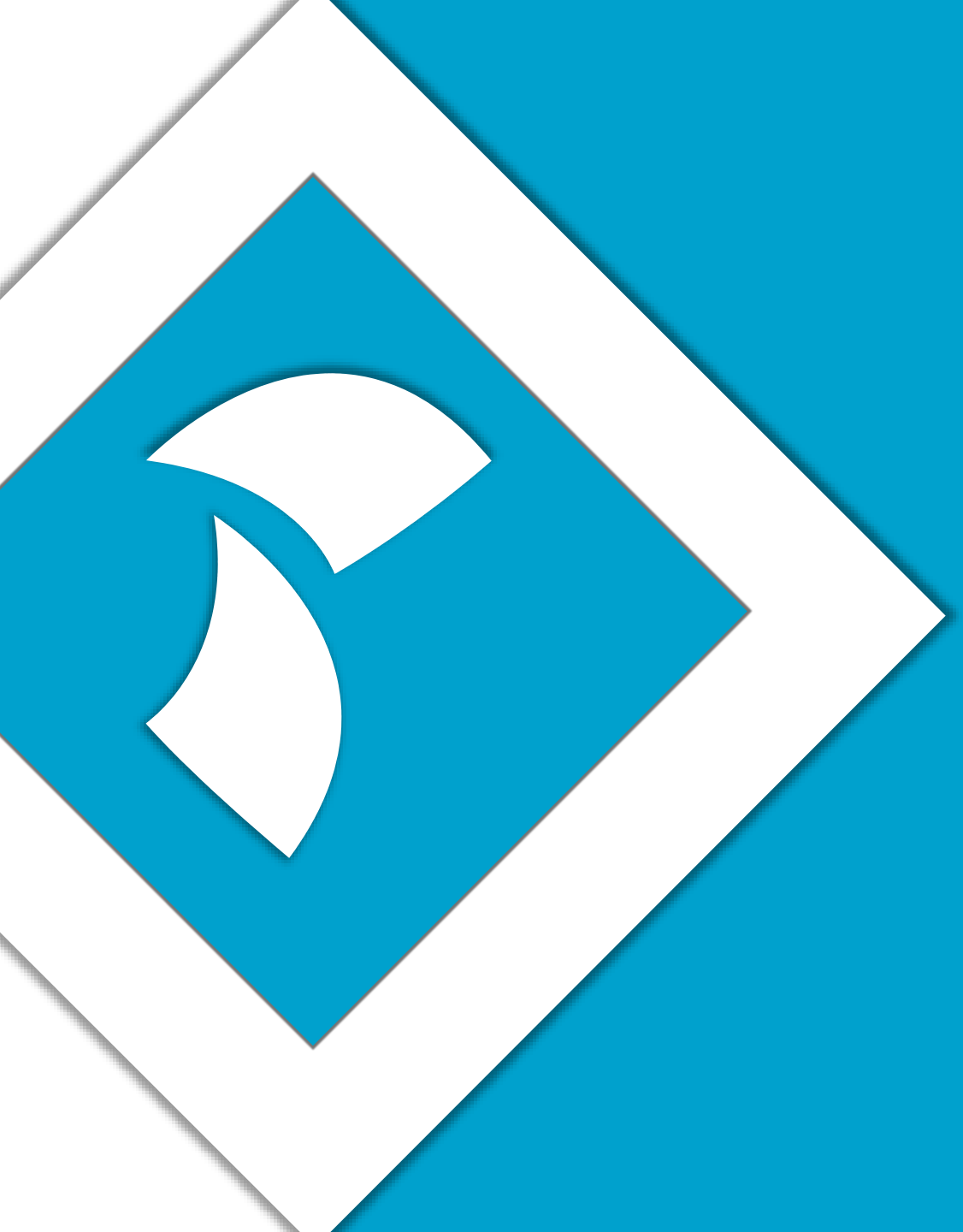


Instruction Explorer

- Gives an overview of all applied layout instructions used in the current Layout Set
  - Blue = individual instructions
  - Red = Block or Type instructions
- Useful to quickly see where which instruction is used
  - Especially if it’s automatically applied as they don’t show up as icons
- A collection of all the information from all individual instruction icons (when hovered over) on the left-hand side in the Structure view
  - if you don’t know exactly what you are looking for, hovering over individual instruction icons could take very long to check

Instruction Explorer





Thank you for your time

Gaining **deeper** understanding

www.blaise.com

blaise@cbs.nl

@blaisecbs

@Blaise5