

VisiRule365 (Spring 2022)

VR365 is a browser-based charting tool which lets you draw, save, load and publish charts to the web.

IMPORTANT: – for the purposes of uploading and publishing:

A] for publishing purposes, your chart **MUST** begin with an initial start node named **start** (in lower case)

B] filenames should NOT contain spaces or hyphens Underscores are OK

C] Expressions can contain spaces, but **ONLY IF** they are quoted

On the PC/Windows Desktop, VisiRule Author uses the underlying LPA Prolog for Windows system to help detect syntax errors and link errors and node type errors. In VR365, this is not the case and some errors may not be detected on the browser but will likely generate errors downstream, typically when attempting to publish the chart. This may result in the chart not being published.

Take great care that if you include **spaces** or **hyphens** in expressions, then these expressions are '**quoted**'

VisiRule is also documented in:

- VisiRule User Guide (vsr_ref.pdf)
- VisiRule Tutorial (vsr_tut.pdf)

These can all be downloaded from the LPA web-site:

http://www.lpa.co.uk/dow_doc.htm

In addition, there are multiple annotated charts available on the VisiRule web-site:

<https://www.visirule.co.uk/visirule-charts>

BE VERY CAREFUL WHEN LINKING 2 OBJECTS TO RELEASE ONLY AFTER SEEING VISUAL CONFIRMATION THAT THE LINK IS OK – THIS IS INDICATED BY A GREEN CIRCLE

If in doubt, you can check by :

- select a node and then
- drag it to see that all the (connected) links follows the node.

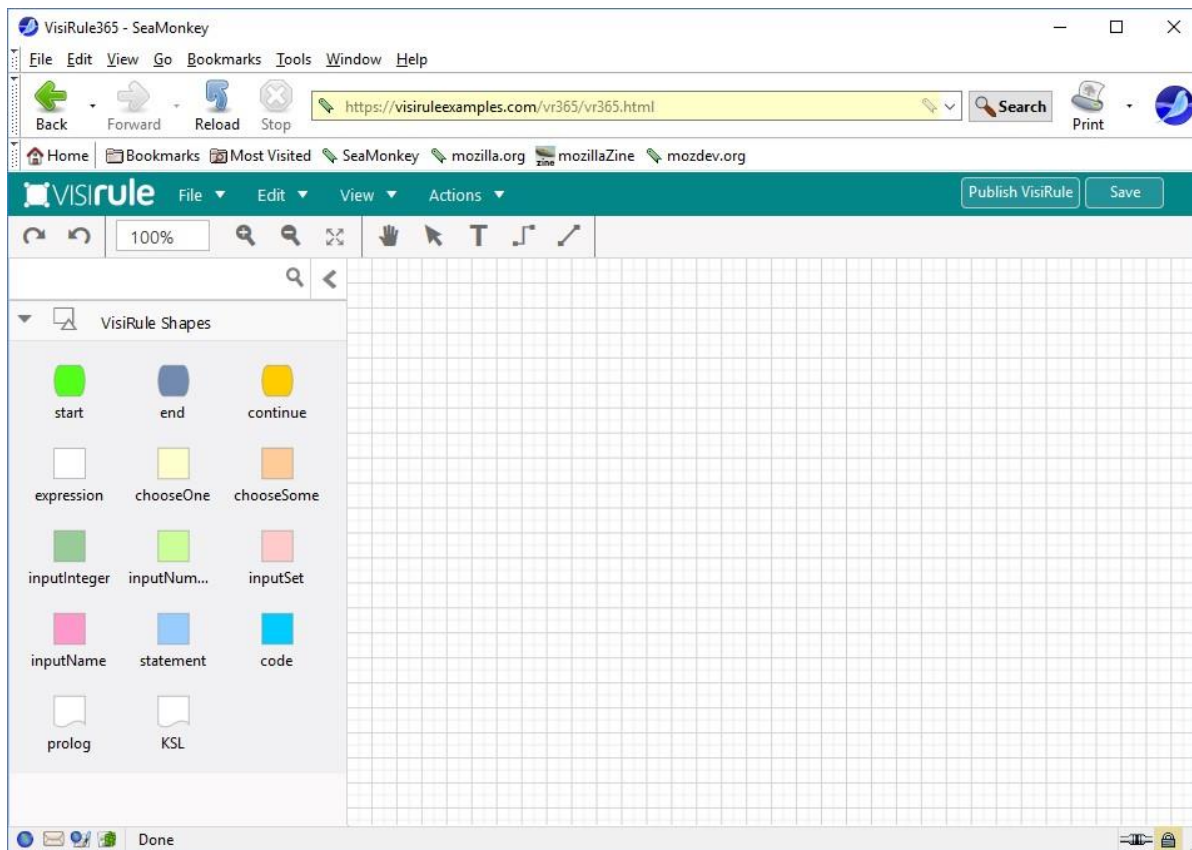
VisiRule will try to inform you of any missing links when you go to Publish the file.

VR365 does generate standard VSR files (which can be used on the desktop), but currently that is a one-way journey.

Contents

| | |
|---|----|
| Initial Screen | 3 |
| Key points in drawing: | 5 |
| Starting a new chart..... | 6 |
| Link the Start node to the Question | 10 |
| Statement Boxes..... | 15 |
| Changing your View | 17 |
| Cloning | 18 |
| VR365 Drop Down Menus | 21 |
| Saving Files..... | 22 |
| How to Publish your Chart..... | 24 |
| Useful Tips..... | 25 |
| The 3 Fields in a Question..... | 25 |
| The 3 Fields in a Statement Box..... | 26 |
| Quotations and Expressions | 26 |
| Quotation Marks..... | 27 |
| Numerical Questions | 27 |

Initial Screen



On the left we see the standard VisiRule Shapes. Each one of these corresponds to a different type of VisiRule Node

Start Node – every chart has at least one of these to start the process/chart

For now – if you wish to publish your chart, it MUST have its main start node named **start**.

End Node – or conclusion – every chart has typically a few of these to end the process/chart – you need at least 1

Continue Node – allows you to break charts into sub-charts; each Continue node MUST match by name with a corresponding start node otherwise you will get an error when this node is reached.

Expression – the value which is used to test the preceding question or statement; acts like a guard in that it has to succeed for the computation to carry on down that branch **chooseOne** – single choice question formed from the values referenced in the expressions following the question

chooseSome – multi-choice multi-answer question formed from the values referenced in the expressions following the question **integer** – integer input question **number** – floating point input question **set** – set input question **name** – text edit input question

statement – a computable question containing executable code which can reference the values in questions

code – a code box which contains explicit Prolog/Flex directives/goals to be immediately executed **Prolog** – can contain any amount of Prolog program statements, routines, functions, predicates

KSL – can contain any amount of Flex program/rule/frame statements or rules, actions, procedures

A typical VisiRule chart has a start node, some questions, some answers in the form of expression logic, and branching, and one or more conclusions.

The diagram of the chart is saved either locally or in the cloud and can then be published.

Local – means the diagram is saved in the user's browser's cache

Cloud – means the diagram is saved in the cloud. It can then be accessed from other machines and from other browsers AND BY OTHER PEOPLE.

If you have obtained your own private VR365 user space, then this is protected from other authors. If you are using the open VR365, then your chart is accessible by others.

You can publish your chart ONLY once you have saved it – either locally or in the cloud.

VisiRule Server will analyse your chart and attempt to generate a Flex KSL program (question, action, group and relations), a Prolog file, an SVG image of the chart and some other data. All of this is accessible using the VR365 Manager.

The generated KSL file is then executed by the VisiRule run-time engine which asks a page of questions, and once the user has submitted their answers, processes the answers to see which expression or expressions have succeeded. It then follows that branch to the next askable question, and creates the next page, until it reaches a conclusion.

If there are problems in the chart, the publish command may fail and result in an error

Key points in drawing:

Drag and Drop Nodes

Drag a node from the Toolbox on the left and drop it on the canvas. You can press and hold your mouse button to select the object, and drag the node to the desired location and drop it by releasing the mouse button.

Edit Text

You can edit the text of a node in one of two ways. Edit text after you create a new shape by immediately typing, or double-click on the node to edit text or select the edit icon.

Connect Nodes

Insert connections between nodes. Just choose one of these methods:

1. Drag and drop one of the line shapes from the panel and select the nodes.
2. Click the port on the right edge of a node and drag your mouse to next node

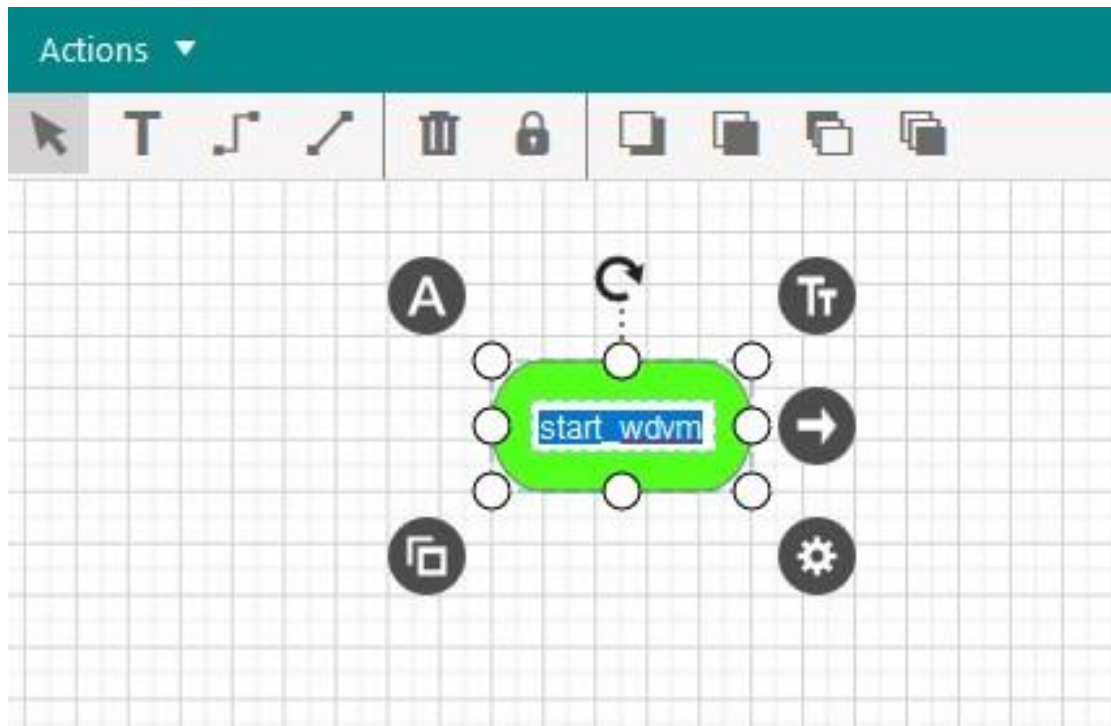
The entire edge of each shape acts as a connection point to attach lines, but its best to connect to ports on nodes. Once you connect a line to a port/node, the line will move with the node.

Take care that you have indeed made the connection – if in doubt try dragging the node. If the connection does not follow, then you need to link again.

Starting a new chart

Let's start by creating an initial Start node.

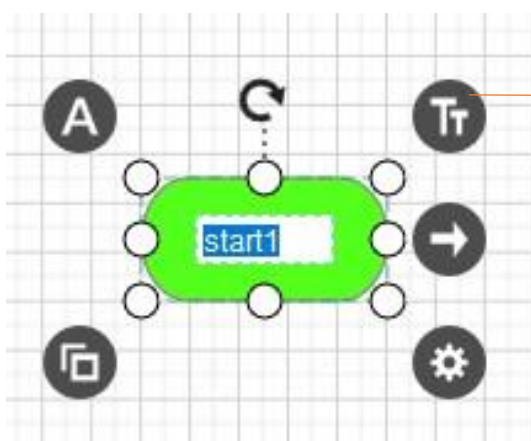
First: select the green shape from the VisiRule Shapes pane and drag it over on to the canvass and release.



Notice we have 8 PORTS --- the white circles - these are important when trying to link nodes together

Notice we have 5 HALOS --- the black circles with white text - these are important when trying to do things

We can edit the node either directly or by selecting the edit halo



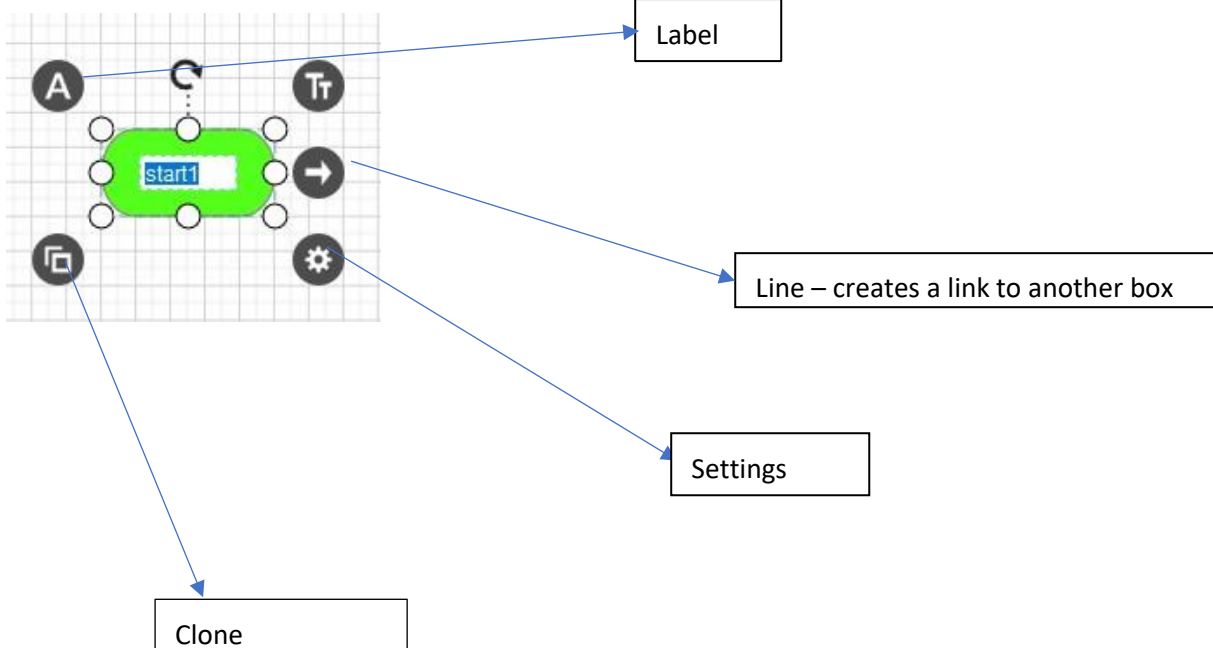
Edit properties in tabbed dialog

Edit Start ✕

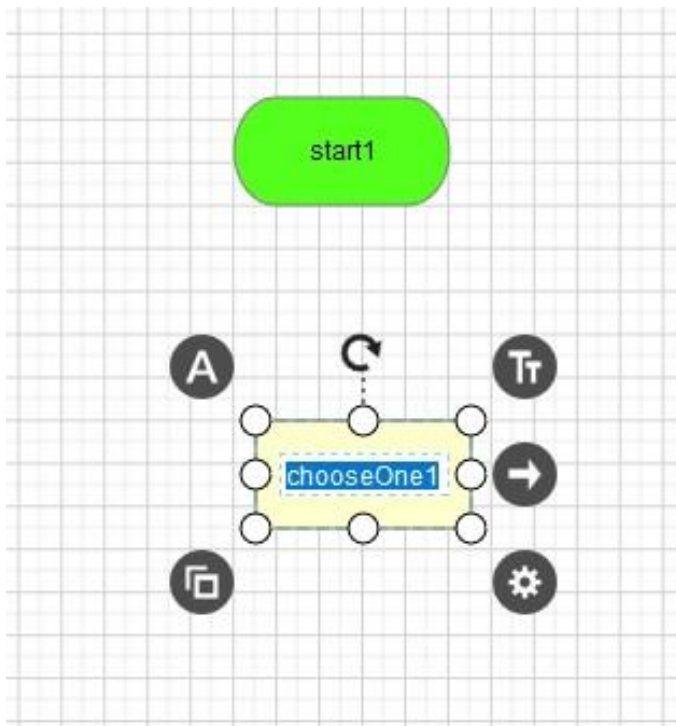
Name

start1

Save



We can add a question:



3 Tabs in Questions

Tab 1]

The Internal Name for the box ... normally not used nor required UNLESS the question is referenced explicitly later in the chart by name within an expression (or statement box).

The Internal Name is mainly visible from a Chart Developers' viewpoint; it is not visible at run-time (other than for debugging and/or reporting purposes); i.e. the user is generally unaware of the names of the questions.

The name is used in the question trail which records and reports on the user's session.

Try NOT to use spaces (or even a CapitalInitialLetter), as this means quotes will be needed in any future expression.

Tab 2]

'The text which is to be presented at run-time as the question prompt'; this is what the user sees when the question is presented.

Tab 3]

An optional piece of text which serves as an explanation could be include a URL which will act as a link ... or any well formed HTML

If interlocutors are supported, such as HOW, WHY, What, then follow the instructions for how to organize them.

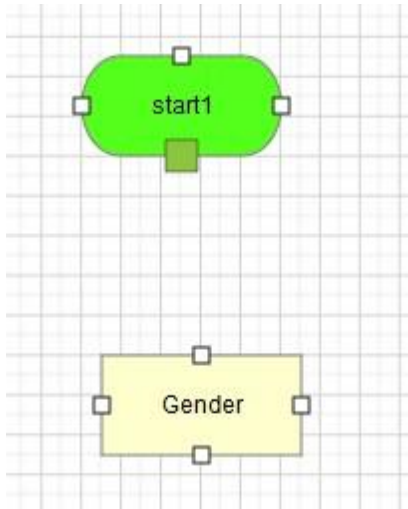
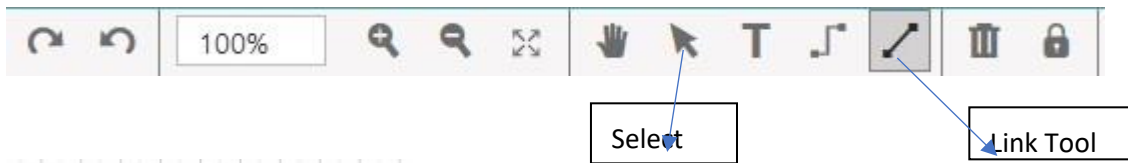
Dialog box titled "Edit ChooseOne" with a close button (X). The dialog contains three tabs: "Name", "Prompt", and "Explanation". The "Name" tab is selected, showing a text input field containing "chooseOne1". A "Save" button is located at the bottom of the dialog.

We can edit the name and separately the prompt tab for the question.

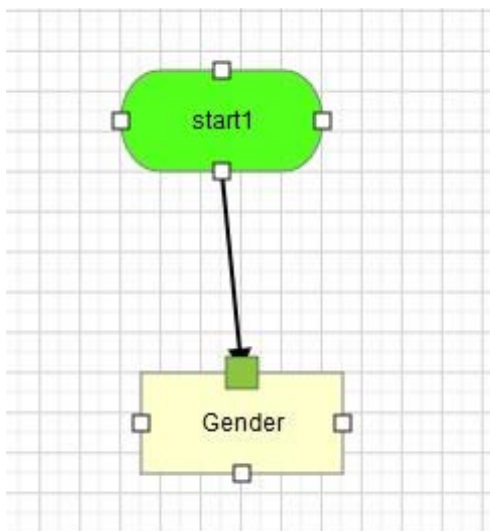
Changing the name of the question does NOT change the prompt, which by default is set to the single word, prompt.

Link the Start node to the Question

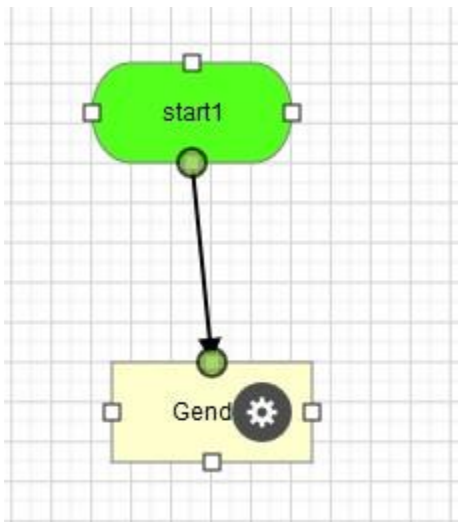
The recommended way to do this is to Select the Link tool from the panel at the top of the canvass



Select a Port on the start node – it becomes a solid green box



Drag to a Port on the 2nd Node – it becomes a solid green box



And release – both Ports should become green circles

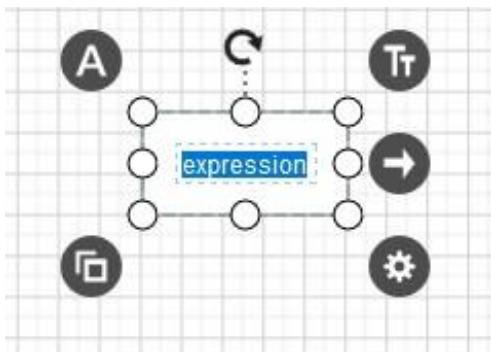
You can check that the boxes are linked correctly, by selecting one of them using the select tool and then dragging the box somewhere. The link should follow!

Conditional logic in VR365 is represented by expressions being linked to Questions.

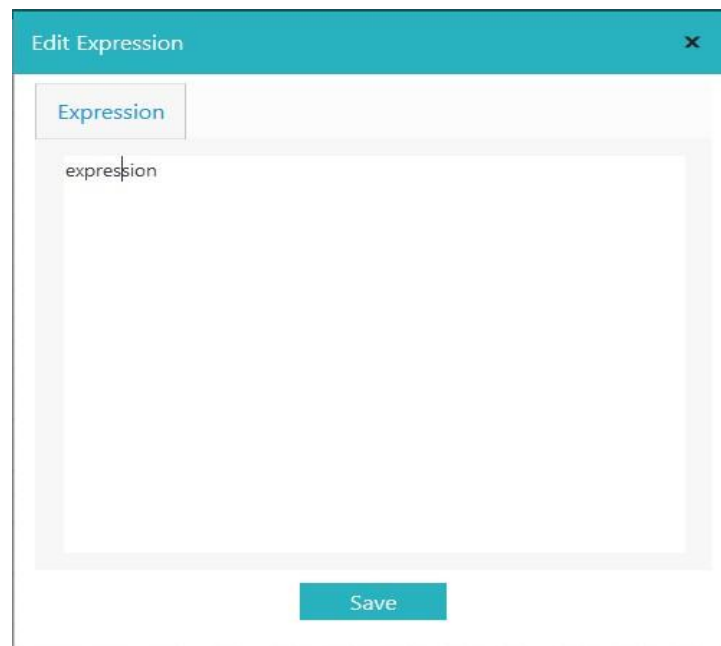
The expressions act as a logical guard; i.e. the expression has to be true for the execution to be able to follow that path i.e. that line of reasoning.

This is the equivalent of an If-Then rule.

We can create expressions using the Expression tool:



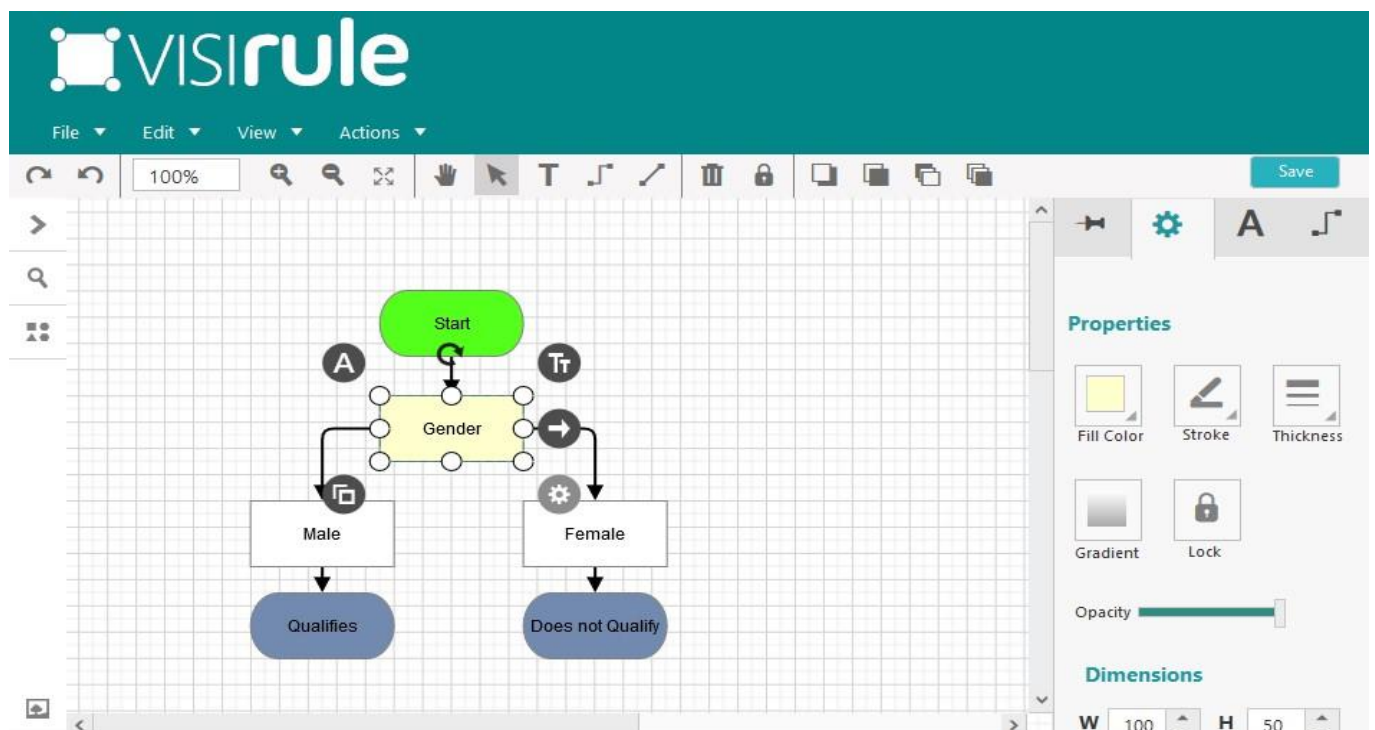
When we edit the expression we only see a single tab field



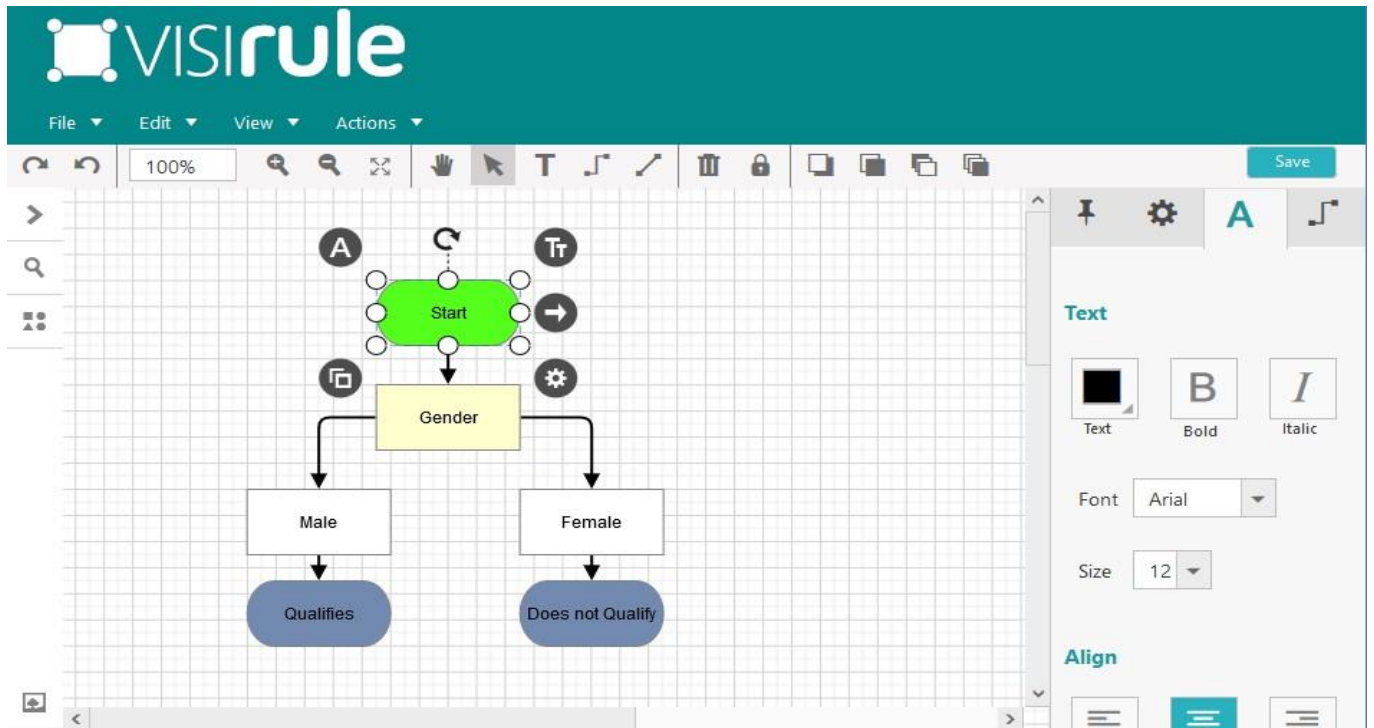
The type of expression we enter should match the type of question we are testing.

So in the case of single-choice questions, we generally test for simple values as the expression.

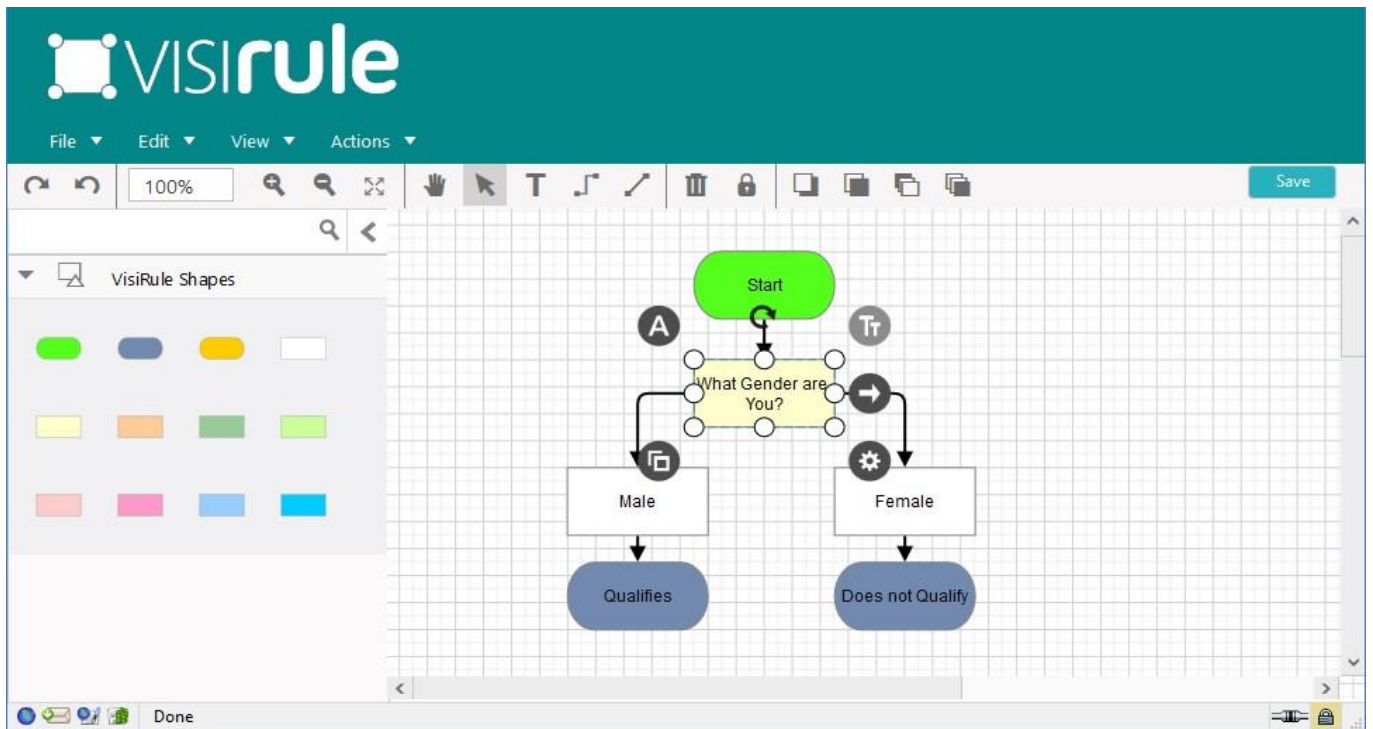
Remember: if you have space in your expression, you MUST quote the expression



The Settings pane provides control over how the node is displayed



The label pane allows us to change the properties of the label

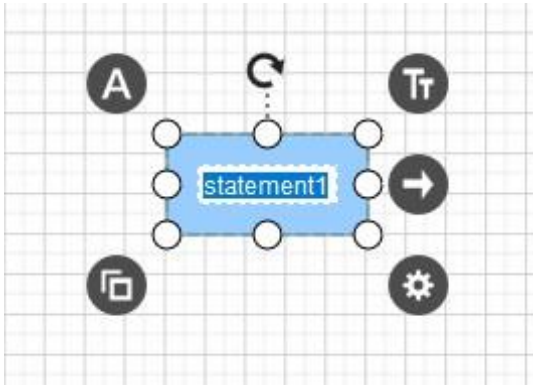


You can also use non-straight lines to link nodes

Again, care must be taken ensure that the link does start ON a Port and end ON a Port

Statement Boxes

Statement boxes are in effect computational questions, i.e. rather than ask the user for an answer, they compute the answer using some code which typically refers to other answers in other questions ... i.e. a derived value based on earlier input or look-up values.



Statement boxes also have 3 fields:

Name

Variable (to be used to expose the result)

Code (which contains at least one use of the exposed variable)

A screenshot of a software window titled 'Edit Statement' with a close button (X) in the top right corner. The window contains three tabs: 'Name' (selected), 'Variable', and 'Code'. The 'Name' tab is active, showing a text input field with the text 'statement1'. At the bottom of the window is a teal 'Save' button.

The 3 Tabs in a Statement Box

Tab 1) The Internal Name for the Statement Box ... normally never used UNLESS the statement box is referenced explicitly by name in a later expression (or statement box) in the chart. As with Question Names, the Internal Name is only visible from a Chart Developers' viewpoint; it is not visible at run-time (other than for debugging and/or reporting purposes)

Tab 2) The name of the logical Variable (UpperCaseInitialLetter) which occurs in the statement box code (Tab 3) which contains the answer

A typical logical variable is:

```
o X o
Answer o
Income
```

Tab 3) Some program code which can use one or more LogicalVariables; only one of which (the one in Field 2) will be used to communicate the computed value of the statement

Logical Implication

Implication syntax (ala Prolog) can be used to conditionally set answers. Examples of implication (If-Then-Else) are:

```
( 'Value' > 0, 'Value' < 50 -> Answer = 'Low';
  'Value' >= 51, 'Value' < 100 -> Answer = 'Medium';
  Answer = 'High' )
```

This reads that is the value of the item called 'Value' is between 0 and 50, then set the Answer variable to 'Low'; otherwise, if it is between 51 and 100, then set the Answer variable to 'Medium', otherwise set the Answer variable to 'High'.

Simple Calculations in Statement Boxes

Examples:

i]

```
Total is 'Question1' + 'Question2'
```

ii]

```
Total1 is 'Q1' + 'Q2' * 12,
Total2 is 'Q3' + 'Q4' * 12,
Diff is Total1 - Total2
```

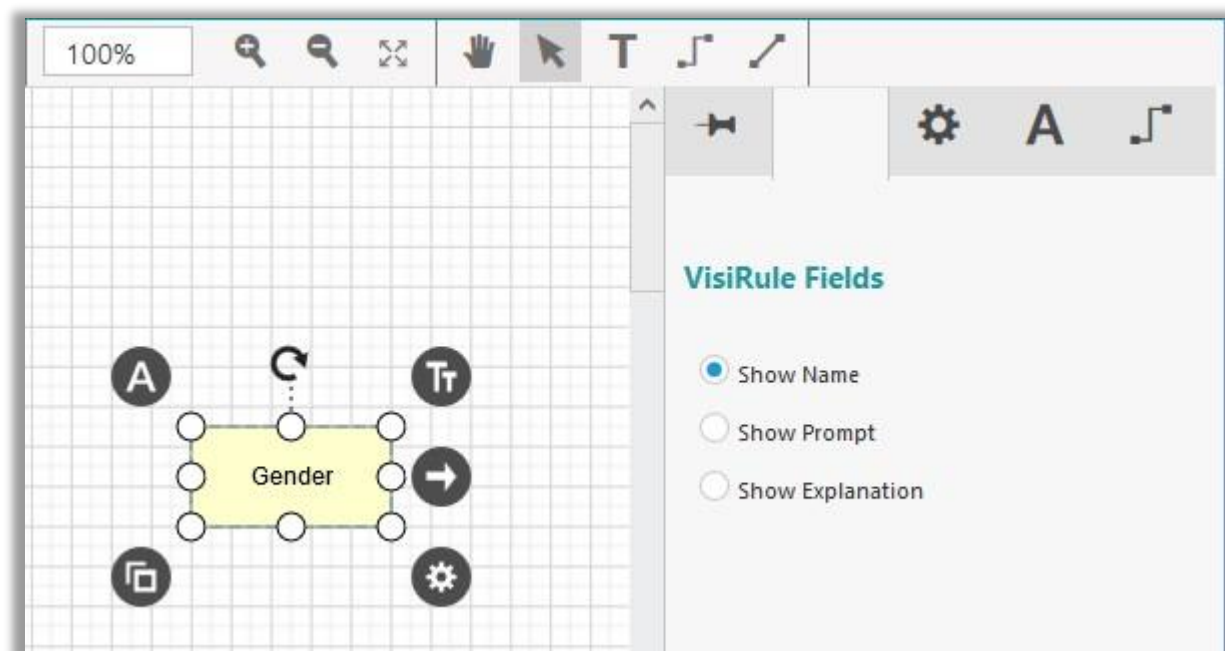
The variable of interest in [i] is Total and in [ii] is Diff

iii]

```
TotalCost is ( 'CarCost' * 1.2 ) + 120
```

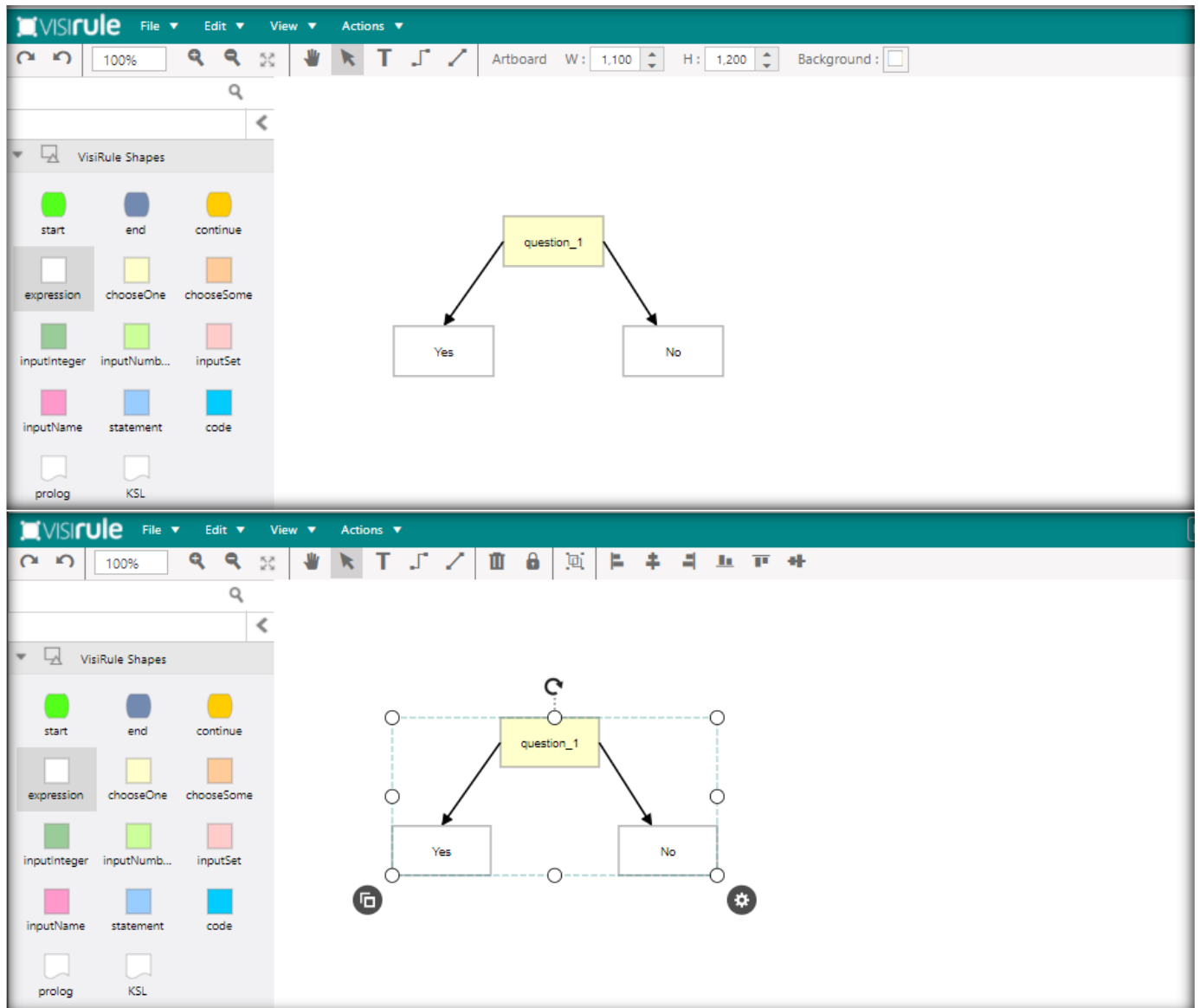

Changing your View

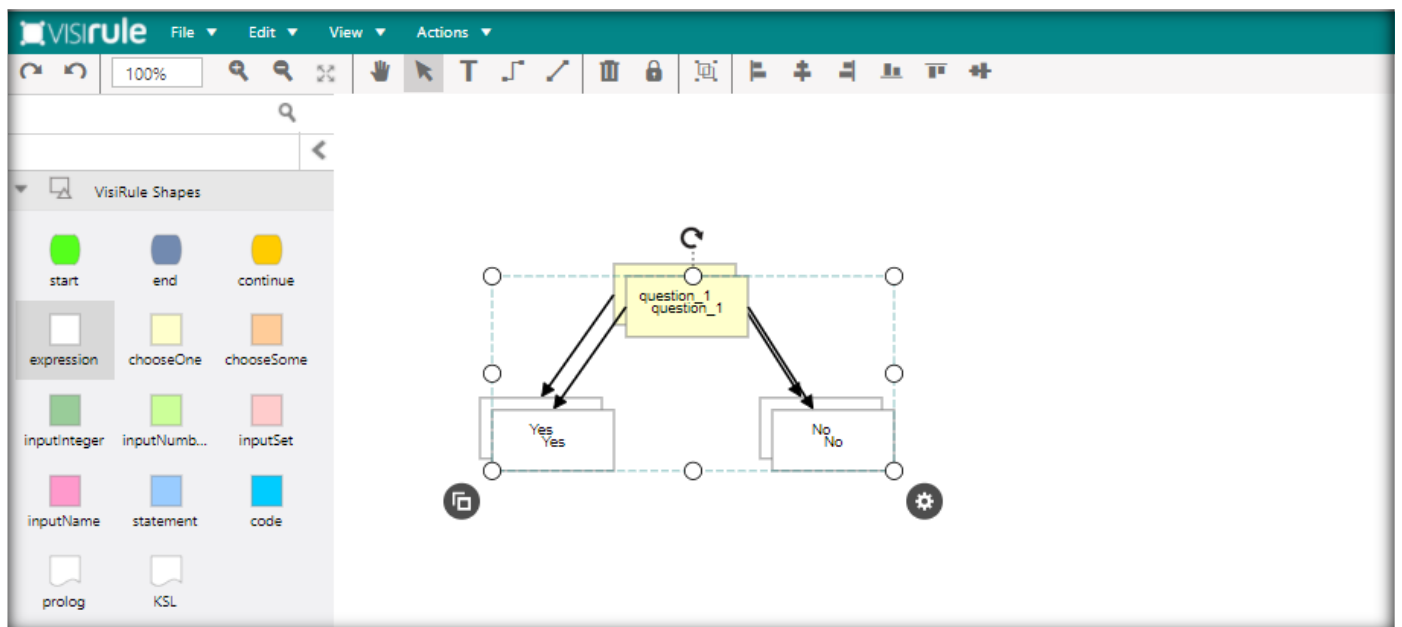
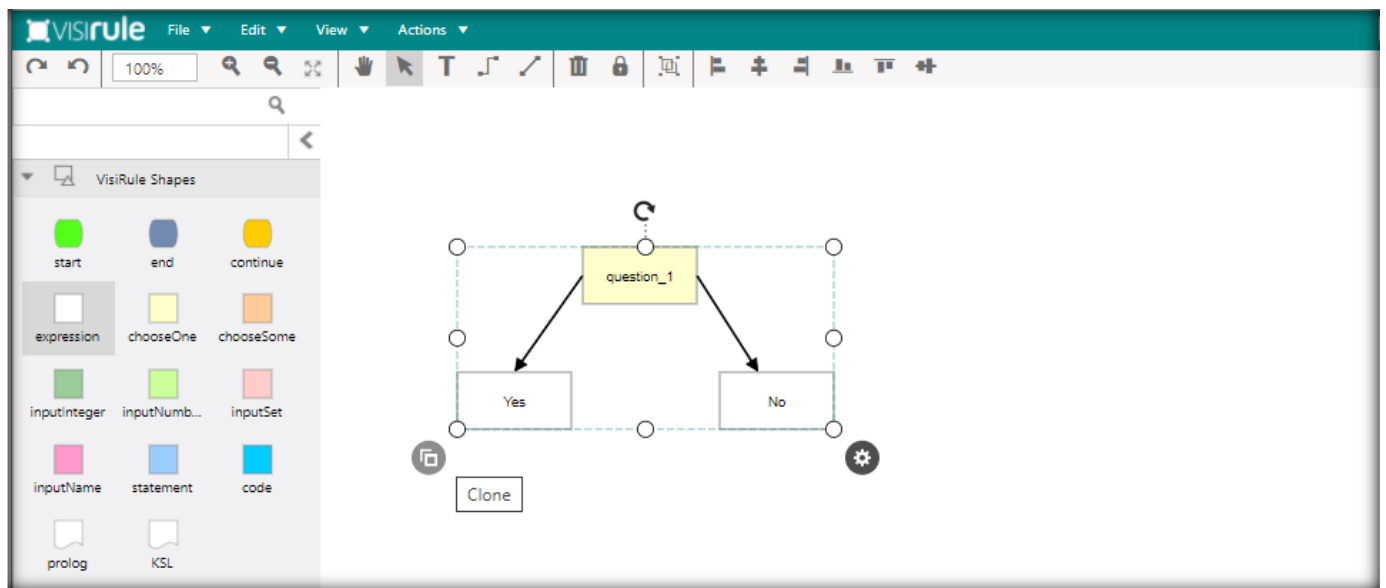
You can change which of the fields is displayed on the canvass for a node.

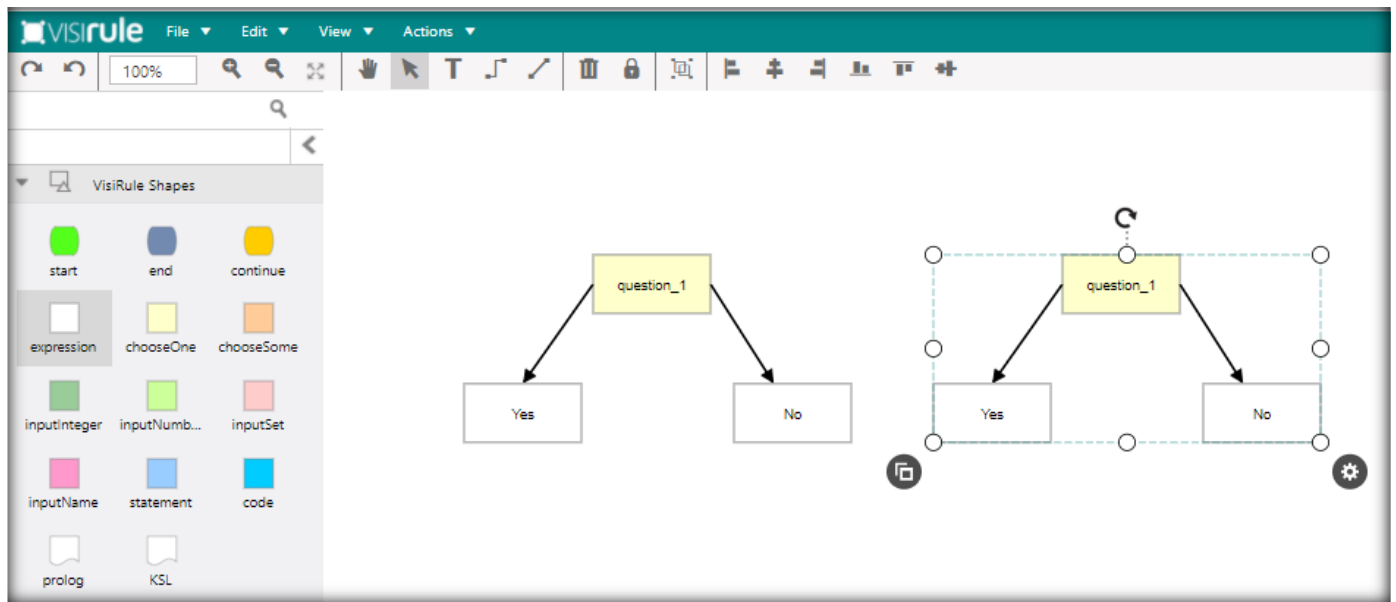


Cloning

Cloning allows you to copy multiple items in a chart and duplicate them.





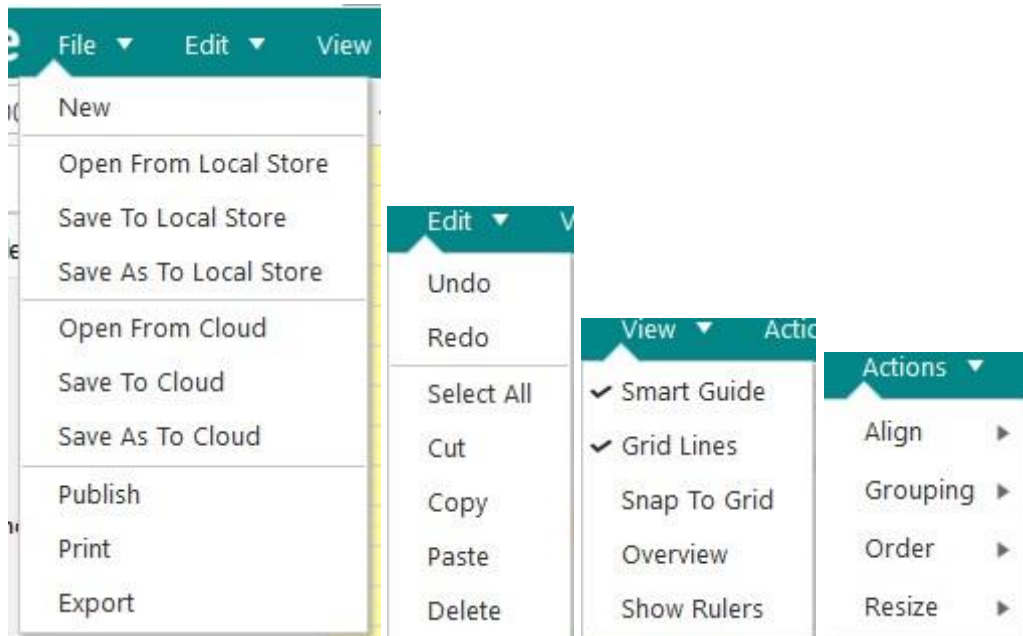


VR365 Drop Down Menus

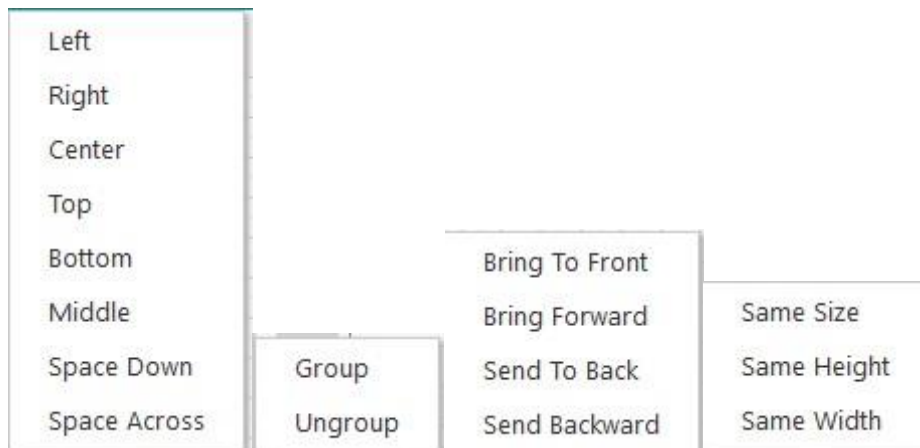
You can drag one or more items by selecting them and dragging

You can also copy/clone one or more items.

There are 4 menus



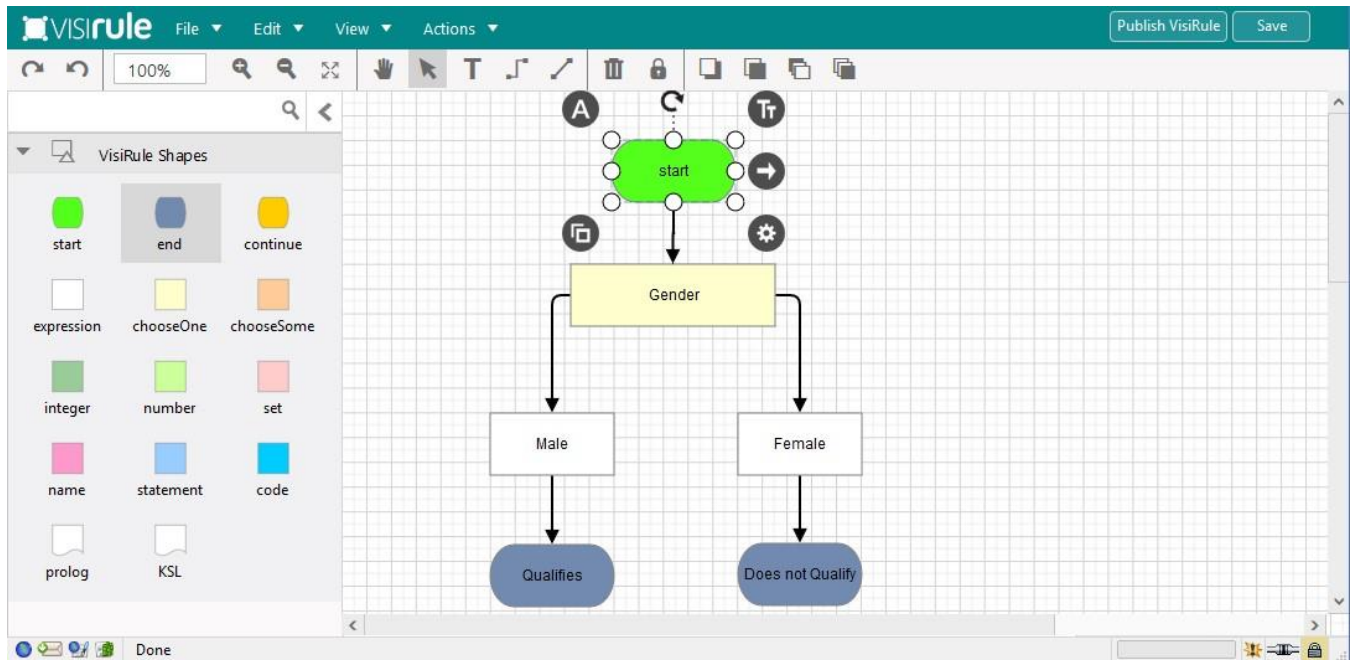
The last menu has 4 sub-menus



Saving Files

You can save your chart, either locally or globally (to the cloud).

Cloud-based saving means you can share file/charts with colleagues. Local saving ensures privacy.



You can save a chart locally so that you can come back another time and carry on editing it on that same browser.

The chart is saved on to your browser.



Do NOT include any spaces in the file name

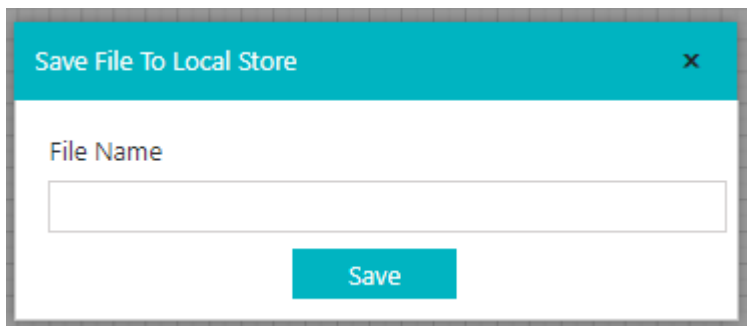


In addition, you can upload and save your chart to VisiRule in the cloud onto the VisiRule Server.

This also allows you to share your chart with other users of VR365.

If you have acquired your own private workspace, then you can save to that and gain privacy.

We can save our chart using the Save to Cloud or Save To Local Store option from the File menu or using the Save button in the Top Right Hand Corner (which saves locally)

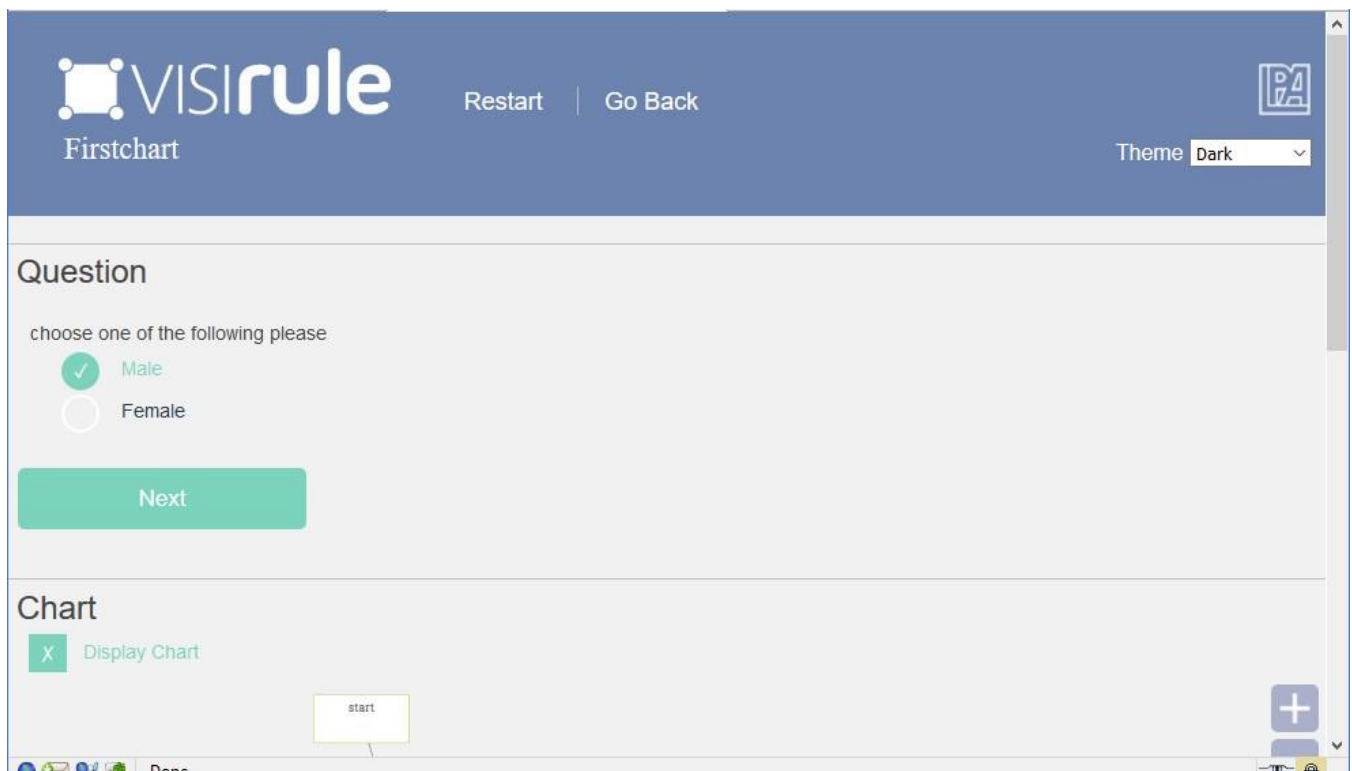
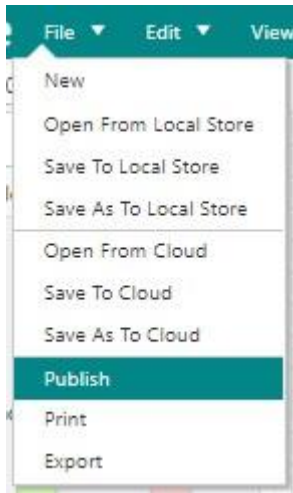


If it's a new untitled chart ... you will be prompted to supply a file name.

Do NOT include any spaces in the file name

How to Publish your Chart

You can Publish your saved chart using the Publish button next to the Save button or by using the Publish option on the File menu.



If this doesn't work then there is a problem with the chart, or the file name. or something else!

IMPORTANT: – for the purposes of uploading and publishing, every chart **MUST** have a main start node named **start**.

Each question needs a unique name.

All nodes need to be properly linked.

Useful Tips

Multiple Choice (i.e. multi-answer) vs Single Choice Questions

Multiple choice questions can invariably be decomposed and then asked as a series of single choice questions. Bit sometimes, you want ask about lots of possible options all at the same time.

Naming Questions

Use MeaningfulNames wherever possible. Each name can only be used once. So, if asking for phone number twice have:

Q1 = HomePhoneNumber or home_phone_number

Q2 = WorkPhoneNumber or work_phone_number

PLEASE NOTE: Questions have to have UNIQUE names

The 3 Fields in a Question

Field 1]

The Internal Name for the box ... normally not used nor required UNLESS the question is referenced explicitly by name within an expression (or statement box) later in the chart;

The Internal Name is mainly visible from a Chart Developers' viewpoint; it is not visible at run-time (other than for debugging and/or reporting purposes)

It is used in the question trail which records and reports on the user's session.

Try NOT to use spaces (or even a CapitalInitialLetter), as this means quotes will be needed in any future expression.

Statements have to have UNIQUE names

Field 2]

'The text which is to be presented at run-time as the question prompt'

Field 3]

An optional piece of text which serves as an explanation could be a URL which will act as a link ... could be a hook into a bespoke explanation system

If interlocutors are supported, such as HOW, WHY, What, then follow the instructions for how to organize them.

The 3 Fields in a Statement Box

Field1) The Internal Name for the box ... normally never used UNLESS the statement box is referenced explicitly by name in a later expression (or statement box) in the chart. As with Question Names, the Internal Name is only visible from a Chart Developers' viewpoint; it is not visible at run-time (other than for debugging and/or reporting purposes)

Field2) The name of the logical Variable (UppercaseInitialLetter) which occurs in the statement box code (Field 3) which contains the answer

A typical logical variable is:

```
X
Answer
Income
```

Field3) Some program code which can use one or more LogicalVariables; only one of which (the one in Field 2) will be used to communicate the computed value of the statement

Examples of implication (If-Then-Else) are:

```
( 'Value' > 0, 'Value' < 50 -> Answer = 'Low';
  'Value' >= 51, 'Value' < 100 -> Answer = 'Medium';
  Answer = 'High' )
```

Calculations in Statement Boxes

some examples:

```
1] Total is 'Question1' + 'Question2'
2] Total1 is 'Q1' + 'Q2' * 12,
   Total2 is 'Q3' + 'Q4' * 12,
   Diff is Total1 - Total2
```

The variable of interest in [1] is Total and in [2] is Diff

```
3] TotalCost is ('CarCost' * 1.2 ) + 120
```

Quotations and Expressions

If you want a single choice question to contain items such as '> 3' and '< 3' then you MUST use quotes in the expression boxes.

Whenever you introduce a space in an expression, you will need to use SINGLE quotations marks later on when referring to that item.

Or you may choose to use underscore:

```
'Cathode Ray Tube'
'cathode ray tube'
cathode_ray_tube
```

Quotation Marks

Do NOT use "double" quotation marks.

Do NOT use `back` quotes

Do NOT use SMART quotes

Use 'single' quotes.

Numerical Questions

It is always good to have at least 2 expressions coming out of a numerical question.

a) tests for $= 0$ and b) tests for some value > 0

This is generally true too for Statement boxes, especially where they are doing some sort of calculation.

End Boxes i.e. conclusions

The node before an end box CAN be (and often is) an expression

If you want to 'continue' your process after the end box has been reached, you can do that but you will need drop down a layer into Flex OR run the chart using a meta-chart.

Incorporating HTML

You can use HTML to enhance the presentation of text on the web simply by using the correct HTML tags.

You can do this for explanations, question prompts and conclusions and within the body of any write or flash statement.

There is an example available which shows how to incorporate text, images, video within VisiRule charts.

Debugging

A very useful ploy is to include (code boxes with) flash/1 or write/1 calls; i.e.

```
flash( 'CurrentValue' )  
write( 'We have reached here' )
```

or you can define your own predicate which combines both (flash and write)