


WIN-
PROLOG

4.900

**VisiRule
User Guide**

by Rebecca Shalfeld

VisiRule User Guide

The contents of this manual describe the product, VisiRule v1.5, and are believed correct at time of going to press. They do not embody a commitment on the part of Logic Programming Associates (LPA), who may from time to time make changes to the specification of the product, in line with their policy of continual improvement. No part of this manual may be reproduced or transmitted in any form, electronic or mechanical, for any purpose without the prior written agreement of LPA.

Copyright (c) 2008 Logic Programming Associates Ltd. All Rights Reserved

*Logic Programming Associates Ltd
Studio 30
Royal Victoria Patriotic Building
Trinity Road
London
SW18 3SX
England*

phone: +44 (0) 20 8871 2016
fax: +44 (0) 20 8874 0449
email: support@lpa.co.uk
web: <http://www.lpa.co.uk>

WIN-PROLOG and VisiRule are trademarks of Logic Programming Associates Ltd.

17 September, 2008

Contents

VisiRule User Guide	2
Contents	3
Figures	7
About This Document	11
Chapter 1 – Introduction	12
What is VisiRule?.....	12
Intelligent Charting	12
Basic Constructs of VisiRule	13
Chapter 2 – Installation and comments	14
Installing VisiRule	14
Launching VisiRule	14
Important Pragmatics	16
Creating Boxes and Links	16
Box Types	16
Editing Boxes.....	16
Relative Positioning of Boxes.....	16
Box Sizes	17
Box Fields	17
Box Alignment.....	17
Undoing Mistakes.....	17
Editing Charts Offline	17
Chapter 3 – Using VisiRule	18
Opening and running your first Chart	18
Chapter 4 – The User Interface	23
Introduction	23
Creating a New VisiRule Chart	23
The Worksheet.....	24
The VisiRule Toolbar	25
Creating an Object	29
Working with Boxes.....	30
Editing Text	33
Resizing a Box?.....	34
Moving Objects	35
Deleting Objects	36
Linking Objects Together	36
Unlinking Objects	38

Smart Deletion.....	39
Smart Linking.....	41
More Smart Linking	45
Deselecting Objects	47
Navigation	47
Zooming In	47
Zooming Out	47
The Worksheet Menu.....	48
The Box Type Submenu.....	48
The Zoom Submenu.....	49
Show Grid Option.....	49
Snap to Grid Option.....	49
Undo Option.....	49
Redo Option.....	50
Show Code... Option.....	50
The Object Menu.....	51
The Object Menu in Edit Mode.....	51
The Box Type Submenu.....	52
Undo Option.....	52
Redo Option.....	52
Cut Option.....	52
Copy Option.....	53
Paste Option.....	53
Clear Option	53
The Object Menu in Non-Edit Mode	54
Tidying up your Graph using VisiRule tools	55
The Align Submenu.....	55
The Space Submenu	56
Delete Option	57
Properties Option	57
The File Menu	59
New... Option	59
Open... Option	59
Save Option	60
Export... Option	62
The Options Menu.....	65
Show Toolbar Option	65
Show Grid Option.....	65
Snap to Grid Option.....	65
Show All Code... Option.....	66
Show Code... Option	66
Keyboard Shortcuts.....	66
<i>Chapter 5 – The Box Types</i>	68
Introduction	68
Naming a Box	68

Start Box	69
Question Boxes	70
Question Box Types	73
Single Choice Question Box	75
Multiple Choice Question Box	75
Number Input Question Box	75
Integer Input Question Box	75
Set Input Question Box.....	75
Name Input Question Box.....	75
Expression Box	76
Three Components of a Simple Expression	76
Question Component	76
Comparison Operators.....	78
The Value Component.....	82
Combination Operators.....	82
Complex Expressions	83
Layering Expression Boxes	83
Troubleshooting	84
Statement Box.....	85
Code Box	86
A flash Command	86
Using a Question or Statement Name in a Code Box	87
Comment Box	88
Continue Box.....	89
End Box	90
<i>Chapter 6 – Hints and Tips</i>	91
When to Ask Questions	91
<i>Chapter 7 – Executing Your Chart</i>	93
Compiling Your Chart.....	93
The Generated Code Dialog	93
Running Exported KSL Code in Flex	95
<i>Chapter 8 – Errors and Debugging</i>	96
Errors.....	96
Edit Time Errors	96
Syntax Error	96
Code Generation Errors	97
Incomplete Chart	98
Syntax error in code box.....	99
Syntax error in variable field of statement box.....	100
Syntax error in code field of statement box.....	101
Missing Reference.....	102
'Analyse Why?' Flex Dialog.....	103

Changing the Size of the Question Dialog.....	104
The Debugging Process.....	105
Normal KSL Window	105
Displaying a Message or Outputting a Value.....	105
Utilising Syntax Colouring.....	105
Appendix A – Allowed Links.....	106
Appendix B – Initialisation Files.....	107
VISIRULE.CFG	107
PRO386W.INI.....	112
[flex]	112
Index	113

FIGURES

Figure 1 - Launching VisiRule using the VisiRule icon	14
Figure 2 - Launching VisiRule from the Start menu	15
Figure 3 – Initial VisiRule window	18
Figure 4 - Opening a chart	19
Figure 5 - The Hello Chart	20
Figure 6 - Generating the Code	20
Figure 7 - Running the Code	21
Figure 8 - Answer the first question	21
Figure 9 – Reaching a Conclusion	22
Figure 10 - Creating a New Chart	23
Figure 11 - Selecting the File Type.....	24
Figure 12 - An empty chart.....	24
Figure 13 - The VisiRule toolbar.....	25
Figure 14 – Selecting Start Box Tool	26
Figure 15 – Selecting a Box	26
Figure 16 – Pointer Mode.....	26
Figure 17 – Start Box Selected Not in Edit Mode.....	27
Figure 18 – Start Box Selected in Edit Mode	27
Figure 19 - Adding a Start Box	29
Figure 20 - Box Creation.....	30
Figure 21 - Stretching a box.....	30
Figure 22 - Enlarged box	31
Figure 23 - Dragging a box	31

Figure 24 - Dragging a marquee.....	32
Figure 25 - Dragging multiple items.....	35
Figure 26 - Dragging a link.....	36
Figure 27 - Connecting a link	37
Figure 28 - Link warning message	37
Figure 29 - About to delete a link	38
Figure 30 - Smart Deletion before	39
Figure 31 - About to delete.....	39
Figure 32 - Smart Deletion after	40
Figure 33 - Ready to drop onto link	41
Figure 34 - Dragging the box	42
Figure 35 - Drop and split link	43
Figure 36 - Nudge box in line	44
Figure 37 - Three unconnected boxes.....	45
Figure 38 - Link right through them	45
Figure 39 - After creating multiple links	46
Figure 40 - Background popup menu.....	48
Figure 41 - Box Type popup menu	48
Figure 42 - Zoom popup menu.....	49
Figure 43 - About to change box type	51
Figure 44 - Box Type Submenu	52
Figure 45 - Object popup menu	54
Figure 46 - Align popup menu.....	55
Figure 47 - Space popup menu.....	56
Figure 48 - Properties dialog I	57
Figure 49 - Field Visibility Tab	58

Figure 50 - File Type Dialog.....	59
Figure 51 - Save Dialog	60
Figure 52 - Saving as VisiRule	61
Figure 53 - Exporting as Flex KSL.....	62
Figure 54 - Exporting as WMF.....	63
Figure 55 - Exporting as XML.....	64
Figure 56 - Options menu.....	65
Figure 57 - VisiRule box types.....	68
Figure 58 - Two boxes.....	69
Figure 59 - Two linked boxes	70
Figure 60 - Simple question.....	71
Figure 61 - Answer a question.....	71
Figure 62 - Explanation box.....	72
Figure 63 - Question Boxes	73
Figure 64 - Set expressions	79
Figure 65 - Statement box for Suit.....	85
Figure 66 - Code boxes with Statement boxes.....	86
Figure 67 - Using statement names in code boxes.....	87
Figure 68 - Simple Comment.....	88
Figure 69 - Continuation Box.....	89
Figure 70 - Water Chart 1	91
Figure 71 - Water Chart 2	92
Figure 72 - Generated Code Dialog.....	93
Figure 73 - Generated Code Dialog after compilation	94
Figure 74 - Circular reference error	97
Figure 75 - Missing end box error	98

Figure 76 - Code box Syntax error	99
Figure 77 - Statement box Syntax error for variables	100
Figure 78 - Code box Syntax error in Goal or Code	101
Figure 79 - Missing Reference error	102
Figure 80 - Prolog Flags dialog	103
Figure 81 - Chart for resetting question sizes.....	104

ABOUT THIS DOCUMENT

This document describes how to load, edit, save, compile and run VisiRule charts.

It explains the basic ideas behind VisiRule, concepts such as question, and expressions and how to combine them to produce executable decision graphs.

You should also read the VisiRule Tutorial which is supplied by LPA.

There are also some VisiRule slide presentations available from LPA.

You should also look at the many example charts supplied by LPA – after all, a picture is worth a thousand words.

CHAPTER 1 – INTRODUCTION

What is VisiRule?

VisiRule is a tool for creating decision support software by drawing charts.

VisiRule is a graphical tool which lets you draw questions and expressions which are mapped into rules. VisiRule generates source code which can be compiled and executed in-situ or exported and used in a separate program.

VisiRule is aimed at people with minimal programming skills who want to model their decision processes. VisiRule enhances productivity by considerably reducing the time it takes to produce a decision support system.

VisiRule combines elements of rule-based processing, knowledge-based processes, knowledge management, decision support tools and artificial intelligence to produce a practical and flexible modeling and problem solving tool.

You can augment VisiRule with code you have written in Prolog/Flex, and/or other languages. This means that VisiRule can be used to model very complex processes.

Intelligent Charting

VisiRule is an intelligent charting tool in two senses.

Firstly, it is used to create knowledge based systems and secondly, it intelligently guides the construction process by constraining what you can and can not do on the basis of the semantic content of the emerging program. This means that, for example, that you cannot inadvertently construct invalid links. As well as this real time semantic checking, VisiRule also checks the syntax of expressions as they are entered, and checks that they are correct with respect to any questions to which they refer.

VisiRule provides the automatic construction of menu dialogs for questions. These are populated by items inferred from expression boxes throughout the chart.

Basic Constructs of VisiRule

VisiRule offers:

- a wide variety of question types including single and multiple choice, numeric and integer entry, text and set entry
- statement boxes which contain executable code for calculating computable values i.e. ones which are not decided by directly questioning the user
- a powerful expression handling logic for testing the values of questions and statement boxes
- code boxes for procedural code and external functions
- continuation boxes which allow you to split charts up into smaller units
- modularity allowing multiple charts and files to define a single executable program

The provision of statement boxes and code boxes, plus the ability to link in conventional Prolog and/or Flex program files, makes VisiRule a very powerful charting system which can do the simple things well and the more complex things too.

CHAPTER 2 – INSTALLATION AND COMMENTS

Installing VisiRule

You should install VisiRule using the LPA CD (or download) as per the chapter in the **WIN-PROLOG** User Guide.

Launching VisiRule

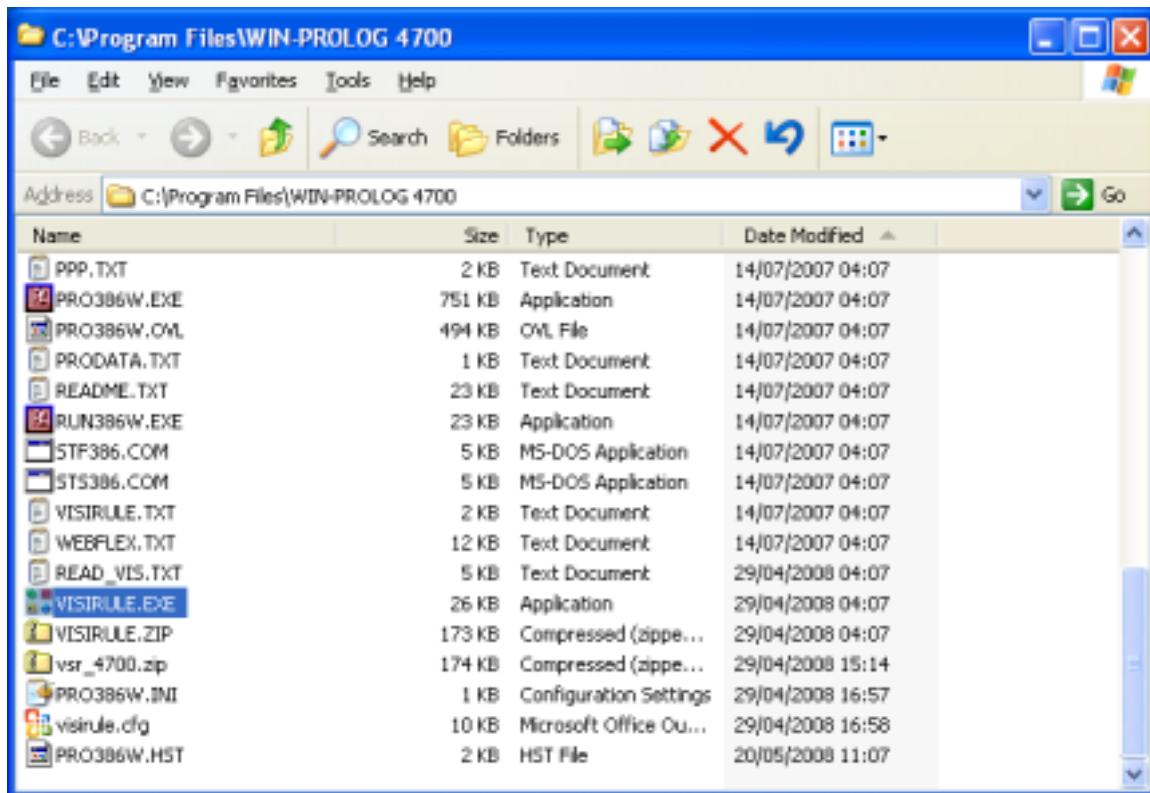


Figure 1 - Launching VisiRule using the VisiRule icon

Once installed, you can launch VisiRule in one of many ways. The easiest way is just to click on the VisiRule.exe icon in the Prolog directory.

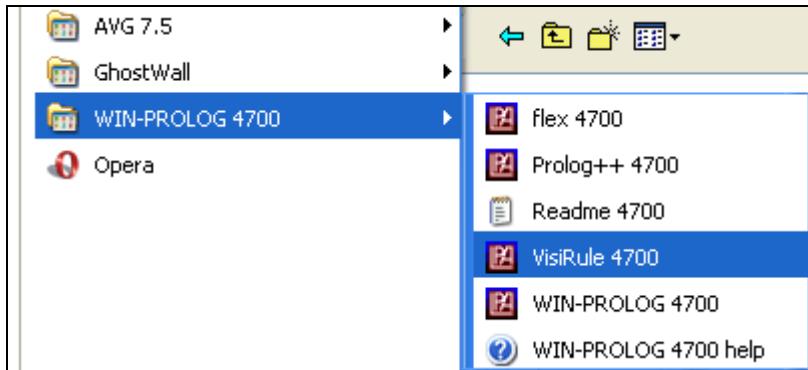


Figure 2 - Launching VisiRule from the Start menu

Alternatively, you can use the shortcut installed on the Prolog Start menu.

For ease-of-use, you can create a VisiRule shortcut on your desktop. Or you can load the VisiRule toolkit explicitly from within Prolog and/or Flex.

The first thing you should see is an empty VisiRule window as per Chapter 3.

Important Pragmatics

A VisiRule chart is composed of an arbitrary number of connected boxes or nodes of different types.

There are various steps in drawing a chart:

- creating boxes
- linking boxes
- editing the text of boxes
- aligning and resizing boxes

Finally, you have code generation and the execution thereof to run the chart.

Creating Boxes and Links

The order in which you create the boxes and links is not significant. If you know the structure of the chart, you may want to draw all the boxes first, and then make the connections. In fact, the intelligent linking tool encourages this. Alternatively, you may prefer to link items as you go.

Box Types

You may even find it quicker to create all the boxes in the same mode, and then set them to the correct box type using the Right-Click option. Or you can switch to the correct box type as you are about to create a new box.

Editing Boxes

Boxes do come with a certain amount of default pre-filled text – for instance, questions always come with a numerically generated unique question name, a prompt text called 'prompt', and an explanation text called 'explanation'. You can edit these at whatever stage you like. You may want to get the structure of the chart established before working on the actual wording. You can change the pre-defined prompts using the INI file.

Relative Positioning of Boxes

Whilst the order in which you create boxes is not significant, their physical location is as the VisiRule code generator uses a left to right depth-first search mechanism when collecting menu items to populate menus and also when working out which expression to test in which order. So, if you wanted to implement a chart with multiple solutions and use Prolog's natural backtracking to check for alternate solutions, then the physical layout would influence the order in which the solutions are found.

Box Sizes

You can drag any box to whatever size you like. You can use this to show only some of the text of say a prompt or explanation and keep boxes to the same size. One way of resizing all the boxes to the same size, is to select them all, and then without releasing the mouse, drag them to the minimum size and back out to the desired size.

Box Fields

You can suppress box fields. So for instance, if you do not want to display question names (they are only of internal usage), you can make the first field of any, some or all questions hidden and therefore no longer visible.

Box Alignment

You may choose to leave final alignment of boxes until your chart is established. Prior to that, it does make sense to try and get boxes to line up as this keeps the links vertical and/or horizontal. You are STRONGLY advised to use the right and left arrow keys to nudge selected boxes into line.

Undoing Mistakes

By accident, you may draw a box when you mean to drag a link – if you want remove the box, hit <ESC> to exit the editor and then <CTRL><Z> to Undo the drawing. VisiRule supports multiple levels of Undo thru <CTRL><Z>, but if you are in an active edit field, then you must exit first.

Editing Charts Offline

VisiRule charts are saved on disc as text files. This means that they can be edited offline using standard word processors or text editors. So you can easily make global changes across, say, for instance, all explanations or question prompts.

CHAPTER 3 – USING VISIRULE

Opening and running your first Chart

Let's open and run a simple chart from the "EXAMPLES\VISIRULE" directory.

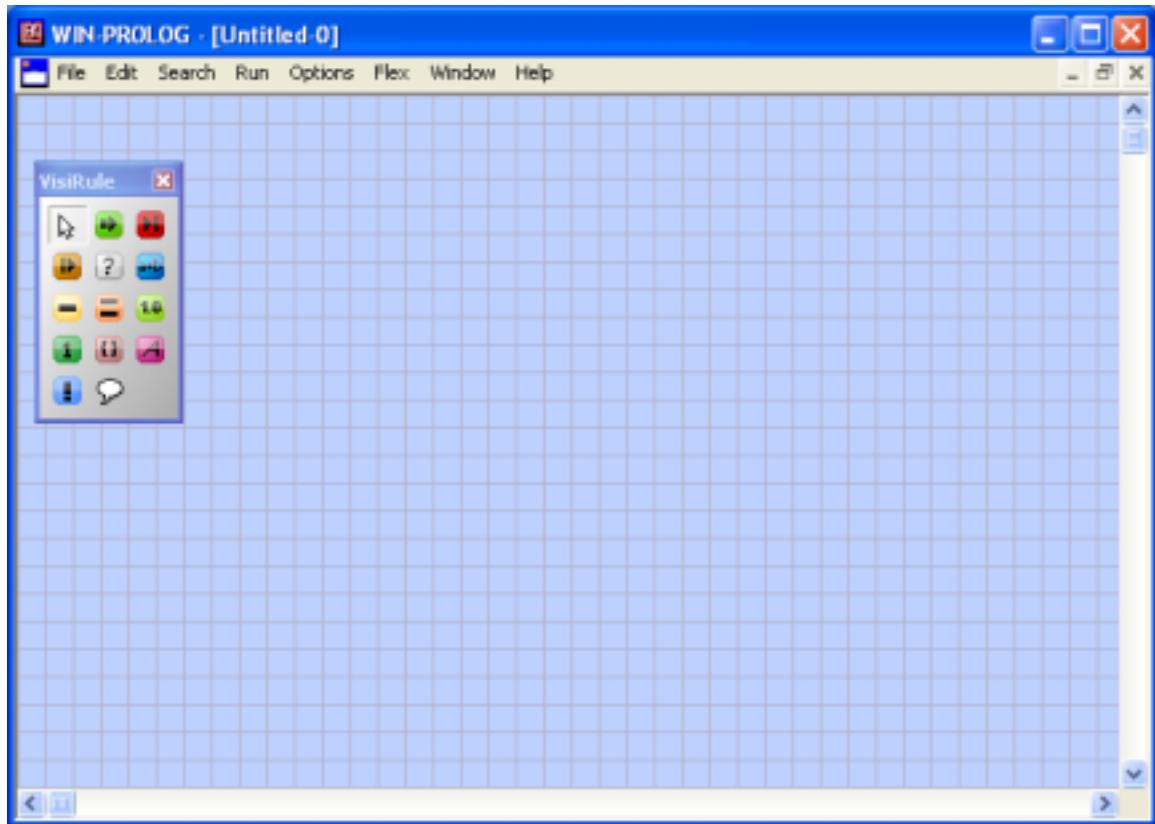


Figure 3 – Initial VisiRule window

Click on the "File\Open" menu option and navigate to the "EXAMPLES\VISIRULE" directory (i.e. go to Examples and then select VisiRule}. Remember to change the "Files of type:" field to "VisiRule Files (*.VSR)".

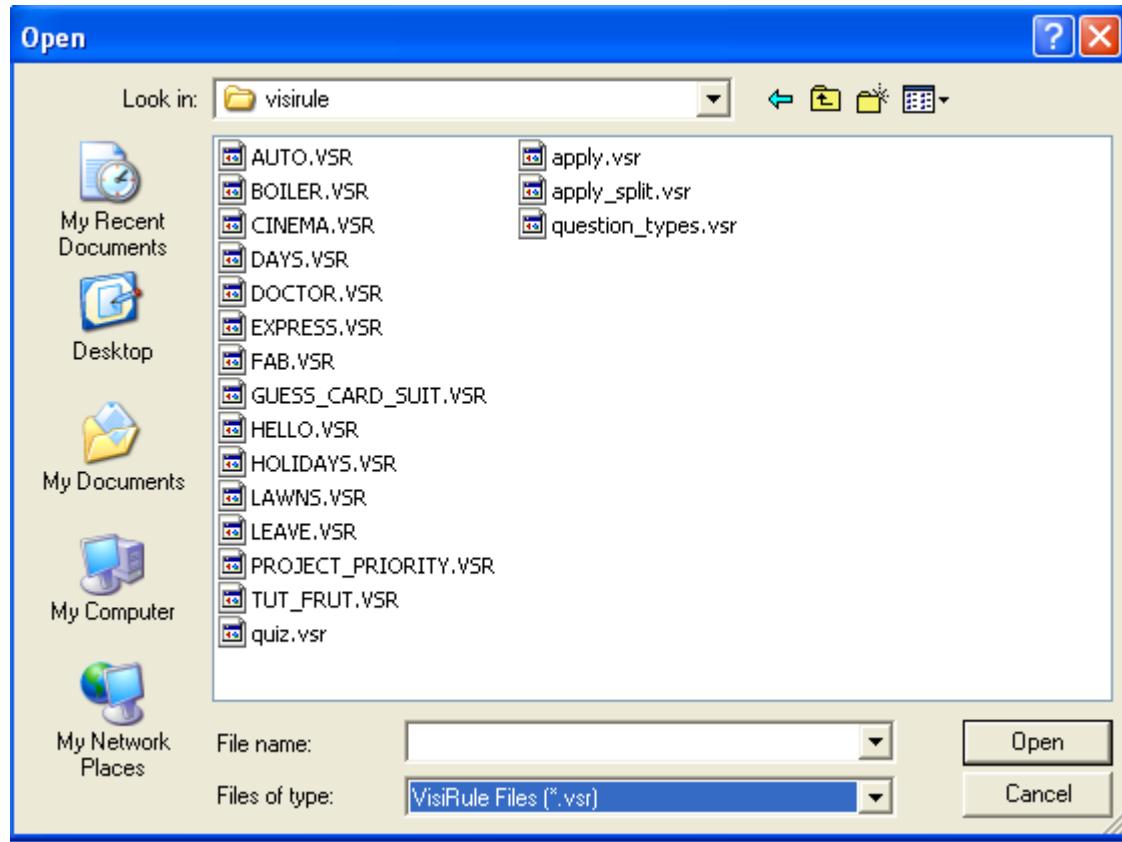


Figure 4 - Opening a chart

Select the "HELLO.VSR" and click on the "Open" button. The "HELLO.VSR" chart will be displayed in its own window:

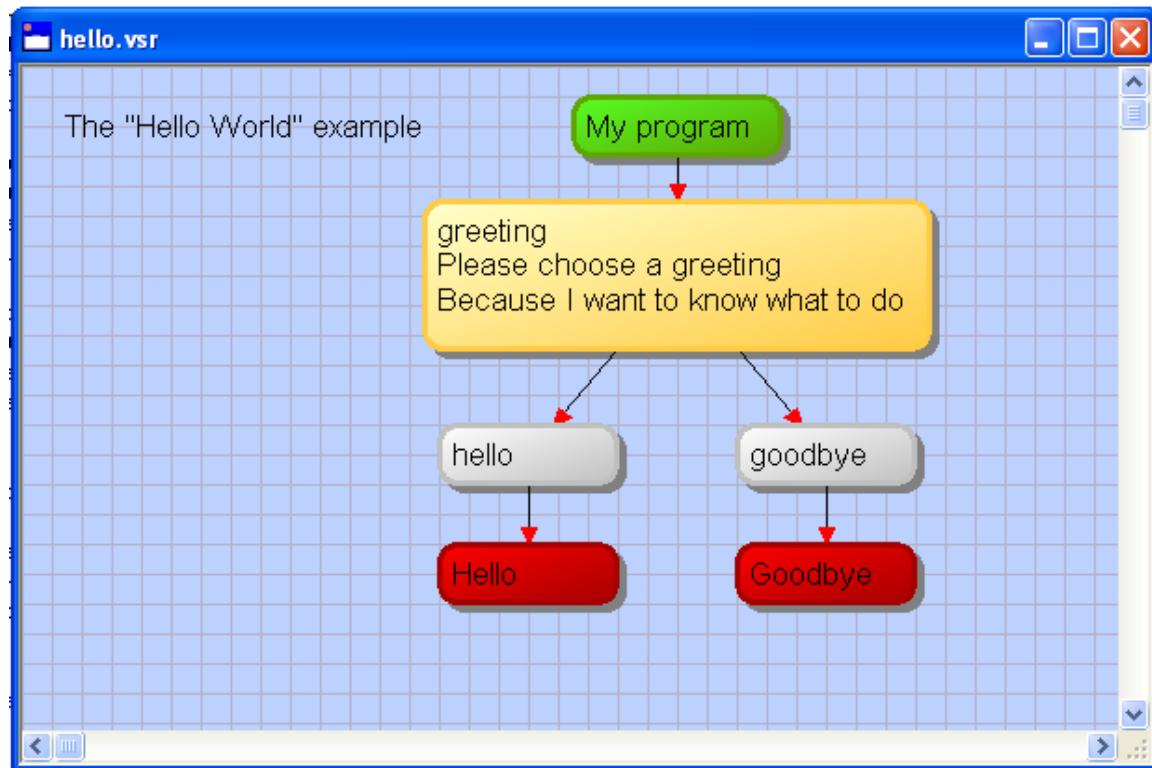


Figure 5 - The Hello Chart

Right-clicking on the background of the chart's window will display a pop-up menu. Select the "Show Code..." menu item at the bottom.

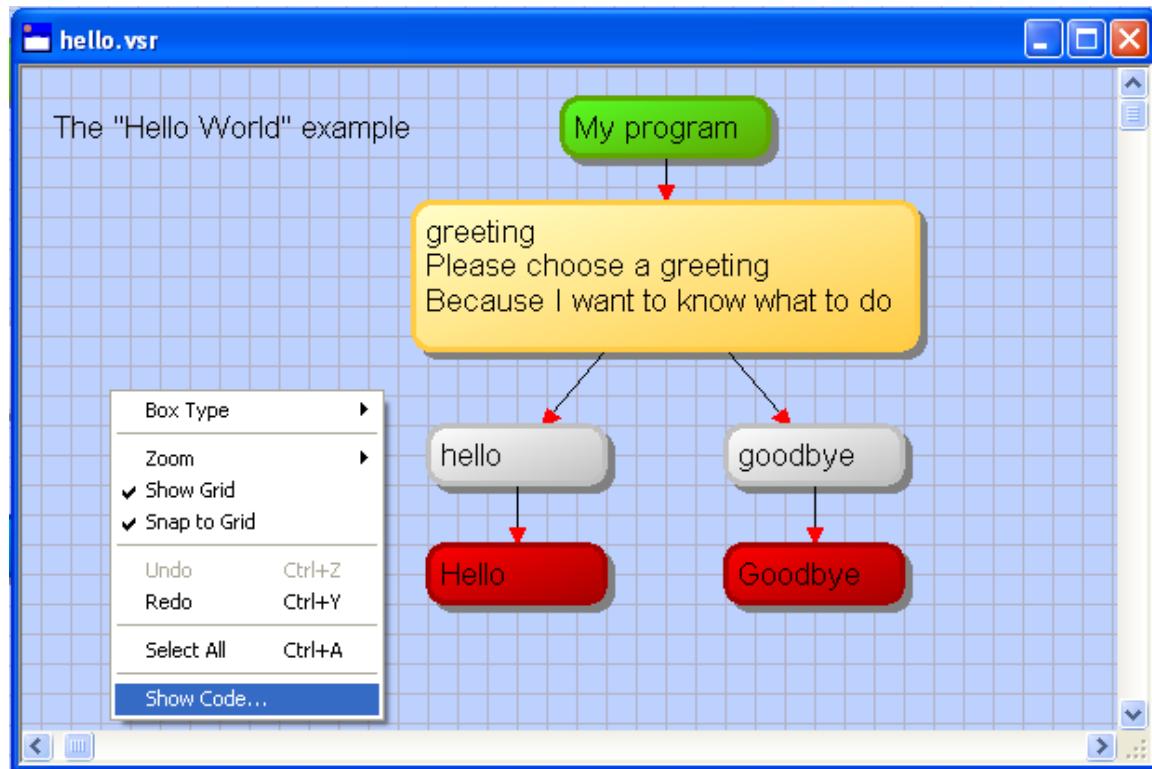


Figure 6 - Generating the Code ...

The "Generated code" dialog will now appear:

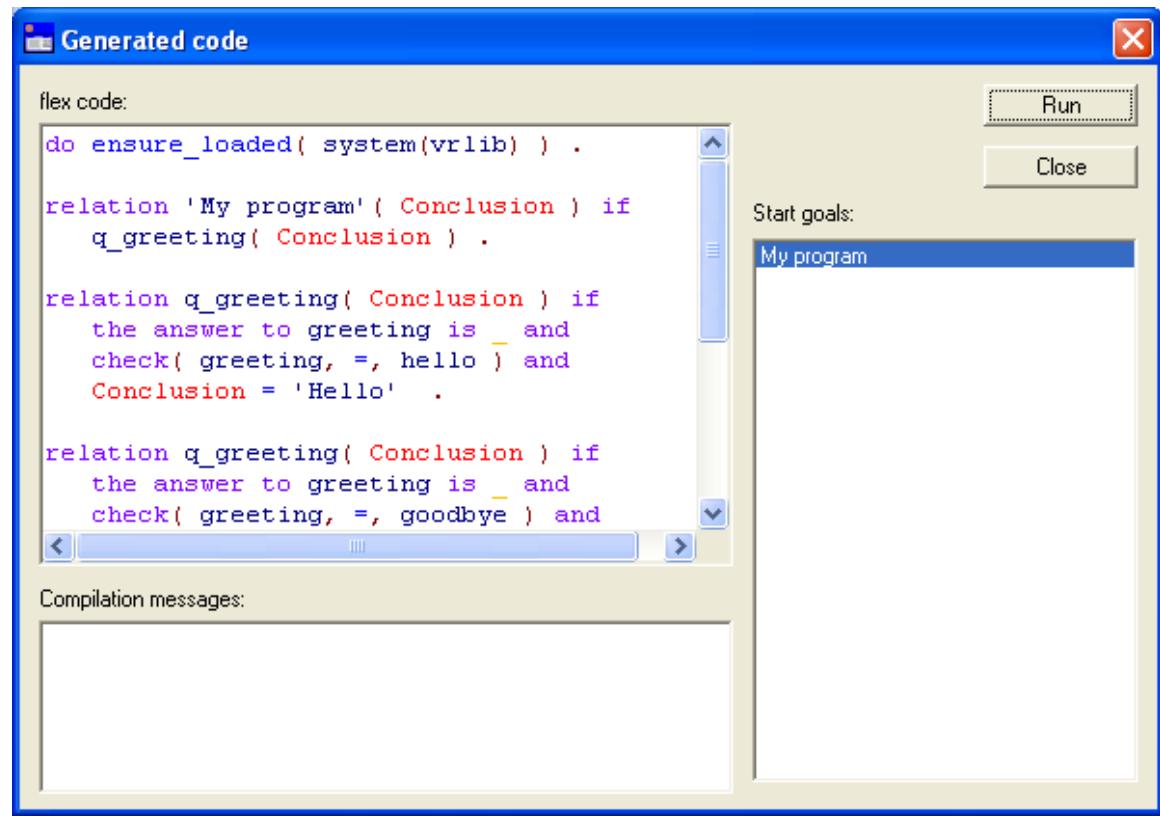


Figure 7 - Running the Code ..

Clicking on the "Run" button will execute the chart.

The first (and only) question will now be asked:

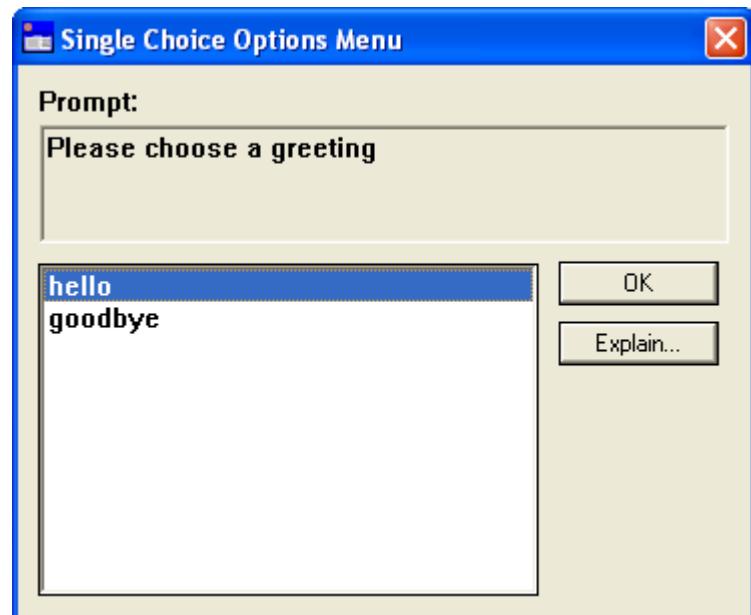


Figure 8 - Answer the first question

Answer the question and a conclusion will be given:



Figure 9 – Reaching a Conclusion

CHAPTER 4 – THE USER INTERFACE

Introduction

The VisiRule user interface consists of three major elements:

- a worksheet for drawing your chart
- a floating toolbar for selecting the current box type and changing mode
- popup menus obtained by right-clicking on the worksheet or on objects in the chart

VisiRule runs within the **WIN-PROLOG** development environment.

Creating a New VisiRule Chart

Click on the 'New...' option on the 'File' menu.

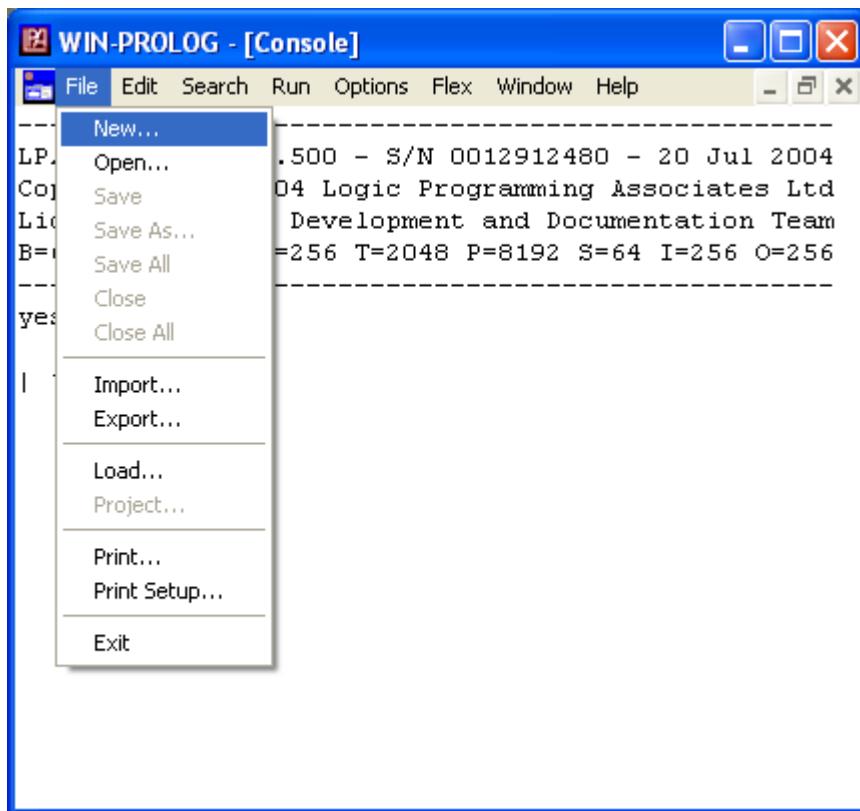


Figure 10 - Creating a New Chart

The New File dialog will appear.

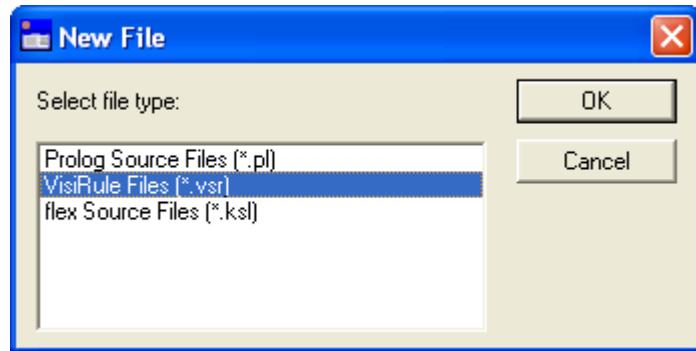


Figure 11 - Selecting the File Type

Select the 'VisiRule Files (*.vsrl)' entry and click on the 'OK' button. A VisiRule window, known as a worksheet, will appear.

The Worksheet

The worksheet is the canvas where you will draw your chart.

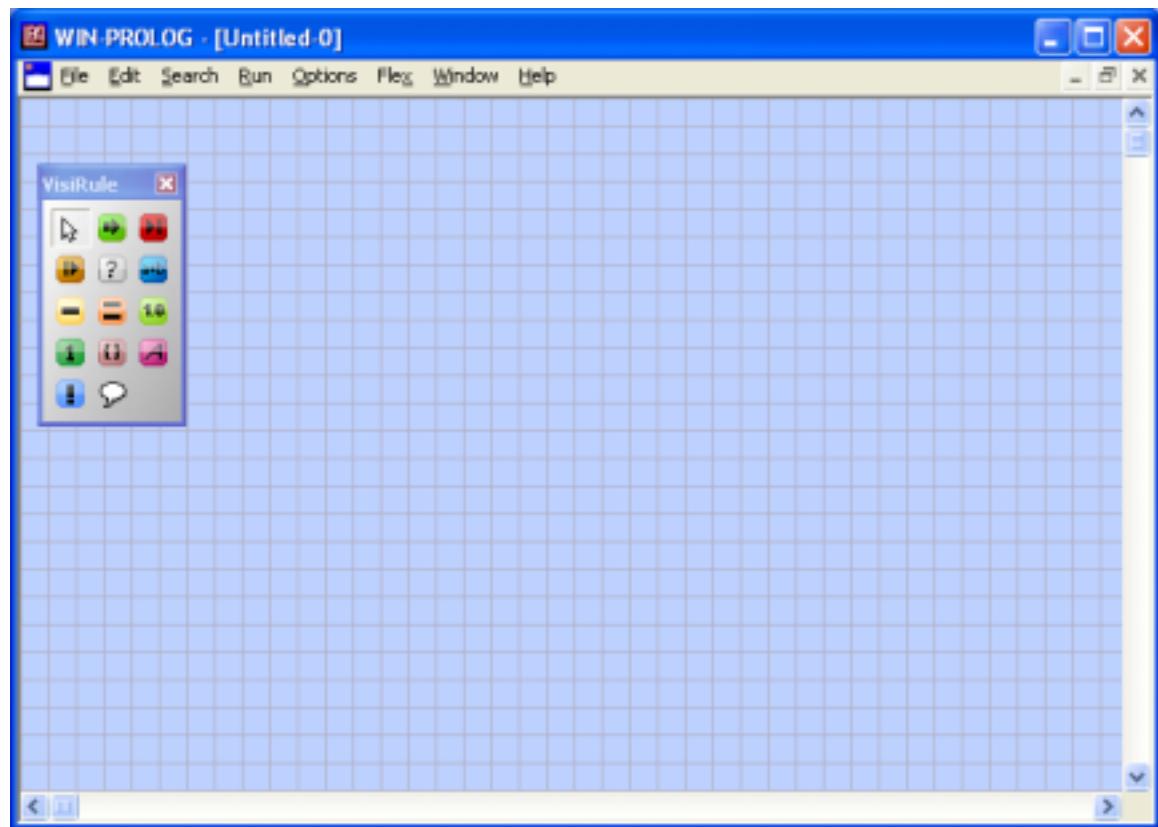


Figure 12 - An empty chart

The VisiRule Toolbar

The VisiRule toolbar is automatically opened along with the first Worksheet window.



Figure 13 - The VisiRule toolbar

The VisiRule toolbar can be shown or hidden using the "Options/Show Toolbar" menu item and can be moved around the screen to any position you desire. The VisiRule toolbar is a global resource, shared by all the VisiRule Worksheet windows.

In Pointer Mode (when the arrow is active) the toolbar allows you to select one or more objects for moving around either by dragging or nudging.

In Create Mode (when anything other than the arrow is active), you can create or edit or link an object.

The current box type is shown as a sunken icon.

You can switch between modes by holding down the CTRL key. This way you can be in create mode, and still select objects for dragging.

Note: in both modes, you can nudge selected objects using the arrow keys.

If you forget which icon is which, hover the mouse pointer over an icon and a tool tip will pop up to indicate what box type it is.

Within the Worksheet, the mouse works in various ways depending on where and how you click, whether or not you are in Pointer mode, and whether or not an edit field is active. Let's click on the green box and select the Start Box tool.



Figure 14 – Selecting Start Box Tool

Now, if we click and drag anywhere, we will create a start box.

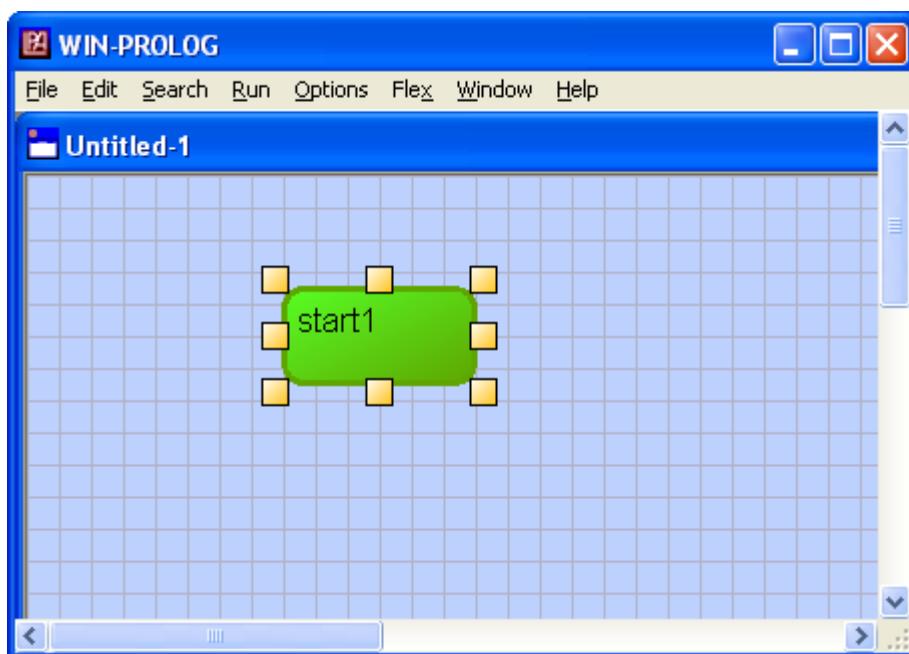


Figure 15 – Selecting a Box

Now if we go back to Pointer Mode.



Figure 16 – Pointer Mode

We can deselect the Start Box by clicking elsewhere in the empty chart.

We can click in the Start Box to reselect it either in edit mode or not depending on where we click.

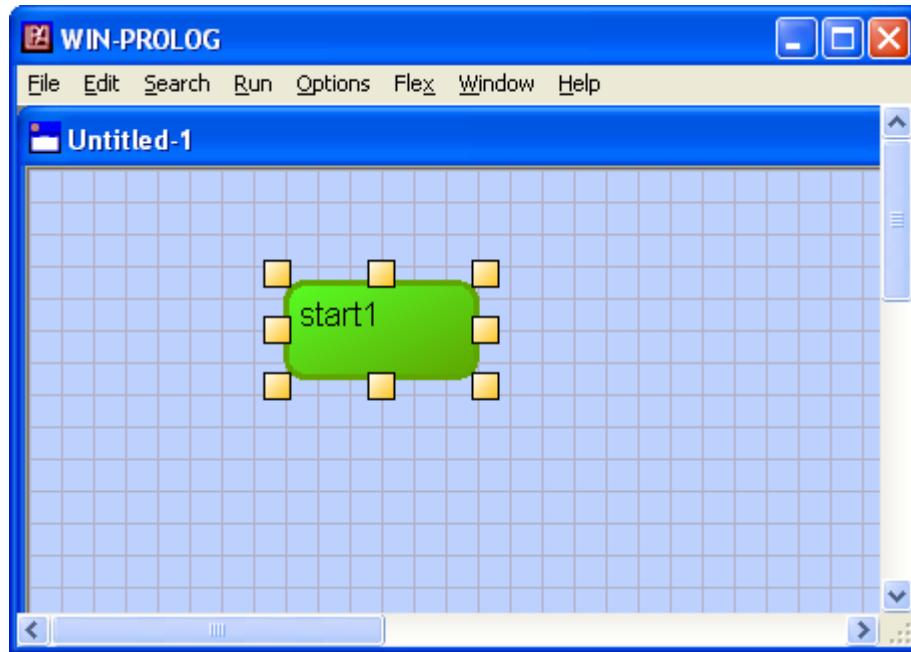


Figure 17 – Start Box Selected Not in Edit Mode

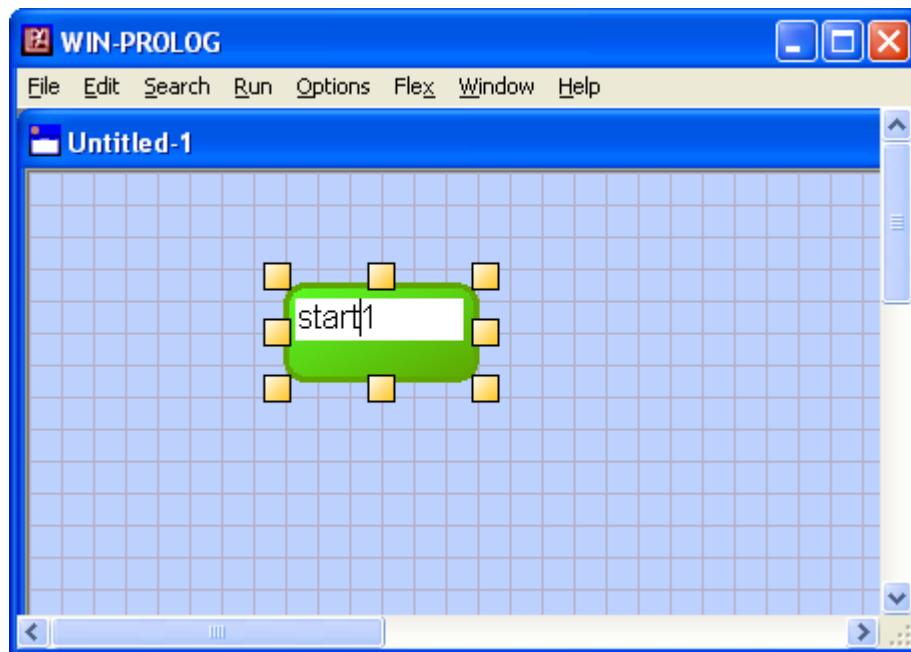


Figure 18 – Start Box Selected in Edit Mode

When not in Pointer Mode, the mouse has multiple functions:

- clicking in empty space, while not in edit mode, will create and select a new object of the current type
- clicking in empty space, with an object is in edit mode, will save any changes made and exit from edit mode
- clicking within an existing object will select it ready for editing
- clicking within an existing object and dragging to another object will attempt to create a link.

You can open as many VisiRule Worksheet windows as memory will permit.

Creating an Object

From the VisiRule toolbar, select the type of box that you want to create, then either double-click or click 'n drag where you want the new box to be positioned.

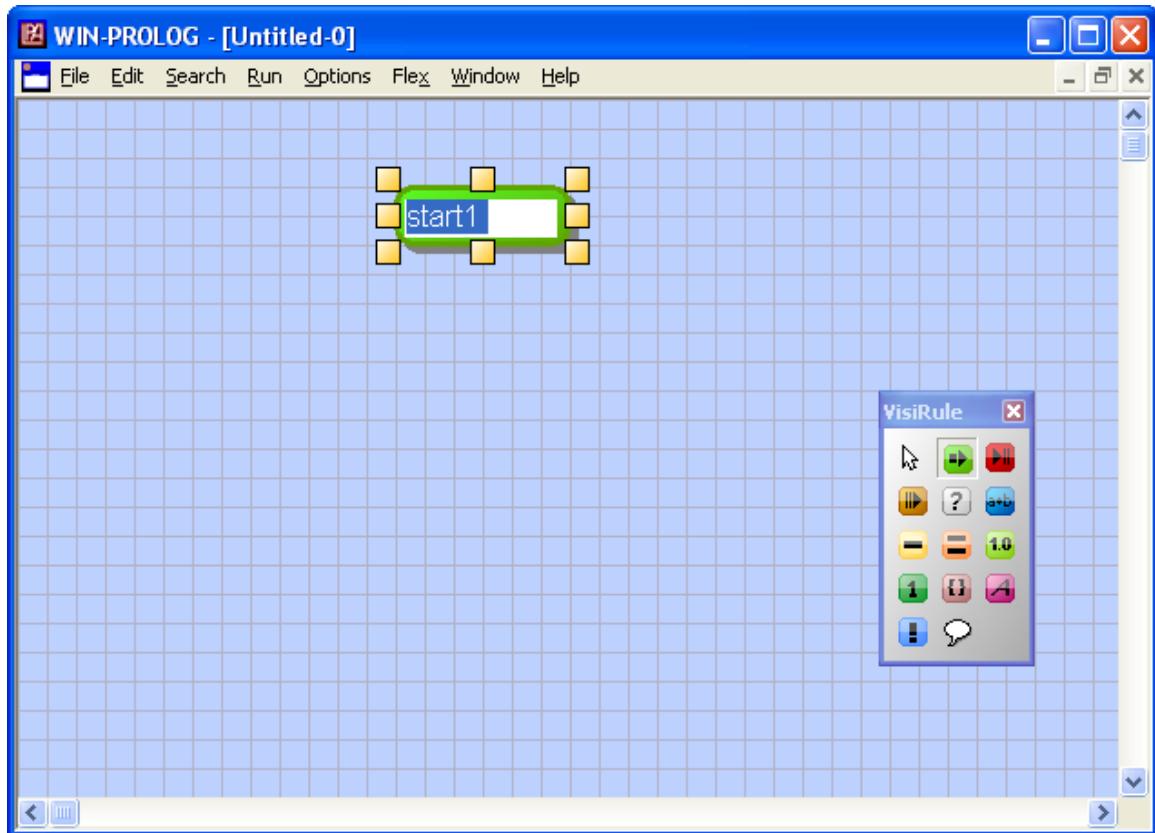


Figure 19 - Adding a Start Box

Immediately after adding the box, it is selected and ready for editing. You are in EDIT mode. This is indicated by the white edit region being shown in the box.

If you do not want to edit the text now, you should hit <ESC> to exit the editor. This will exit you from EDIT mode. The colour of the box will become a solid, single colour

Now, you could, for instance, link this box to another one (if you had one), or holding the <CTRL> key down, you can drag this box to a new location.

When you are in edit mode, all you can do is edit the text.

The yellow handles are for stretching the box, but first you have to get out of edit mode. Then you can resize the selected box or boxes in any direction.

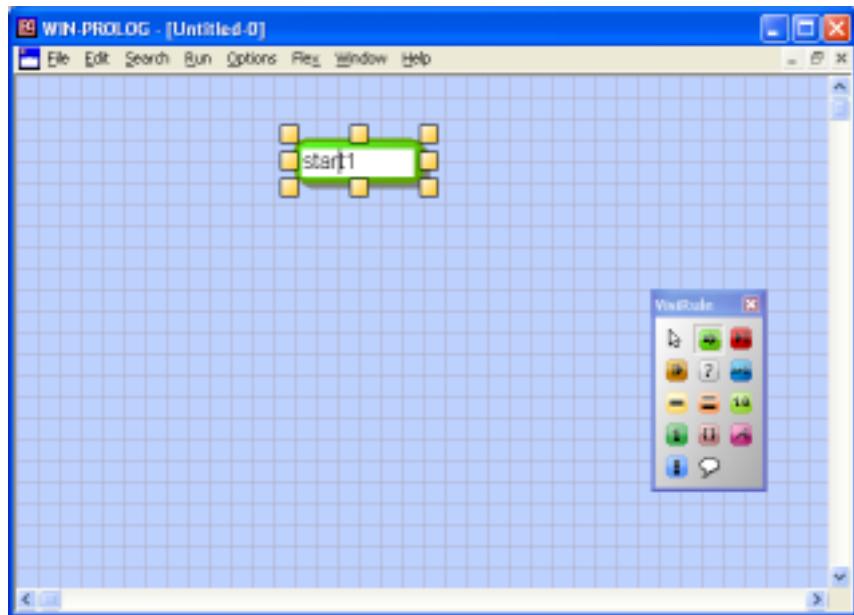
Working with Boxes

There are several ways to work with VisiRule objects:

Creation

The object is in EDIT mode and is ready for editing

Figure 20 - Box Creation



Resizing

Dragging a yellow box indicates wanting to resize

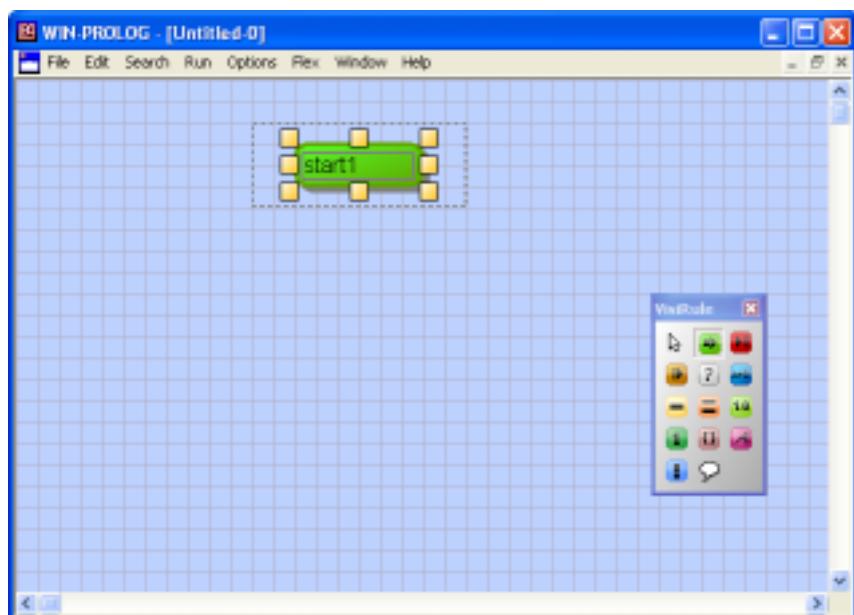


Figure 21 - Stretching a box

The Start box gets bigger

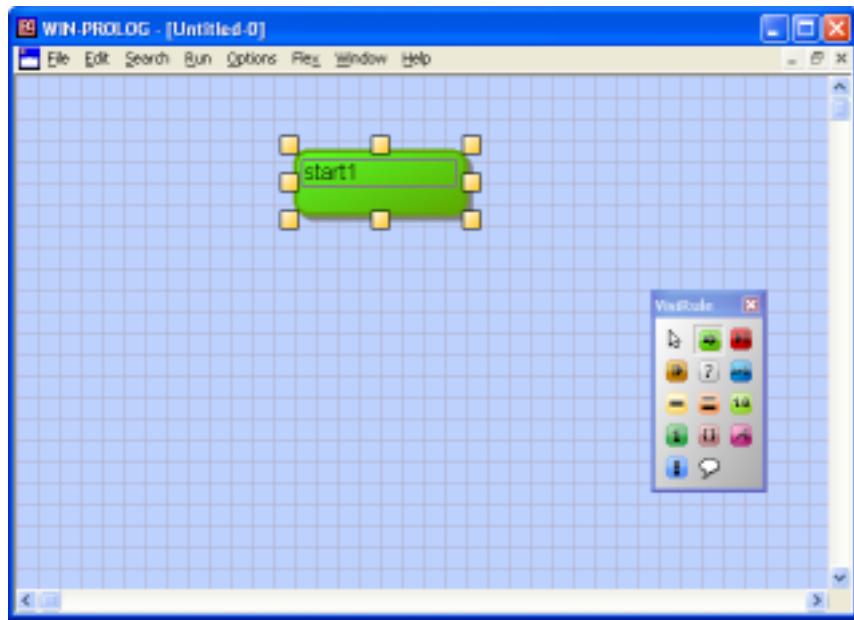


Figure 22 - Enlarged box

Dragging

In Pointer mode we can drag

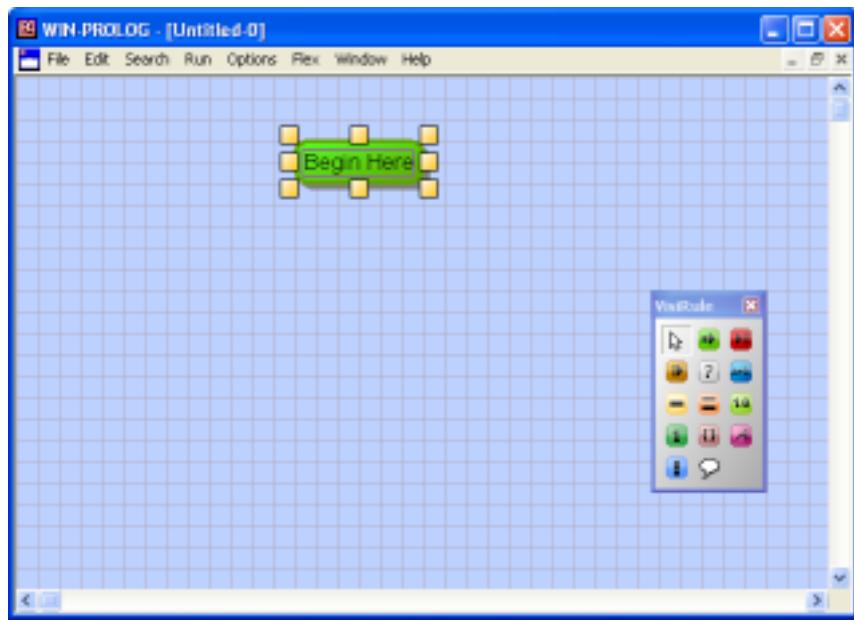


Figure 23 - Dragging a box

Selecting more than one object

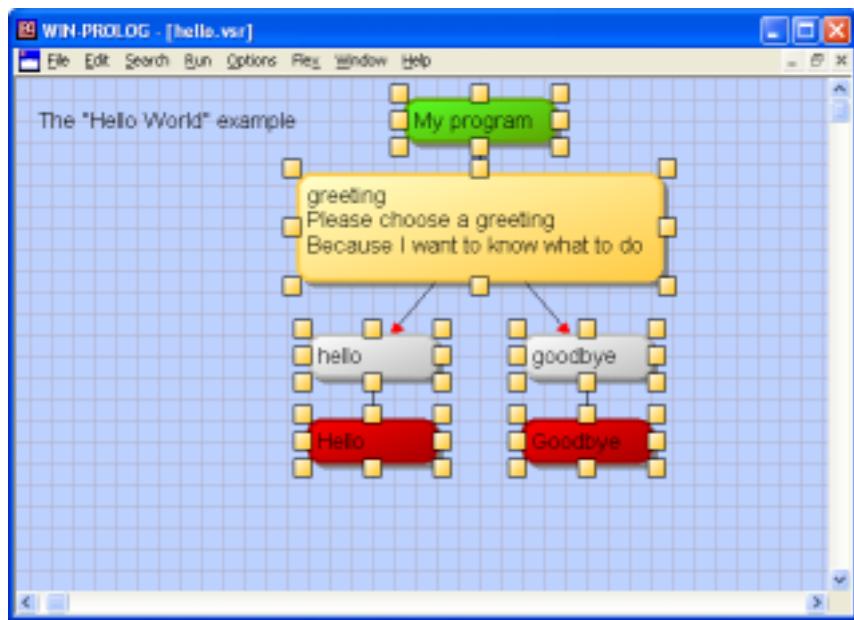


Figure 24 - Dragging a marquee

For an object to be added to the selection, it must be completely within the marquee.

Select all objects Press the <Ctrl> + <A> keyboard combination.

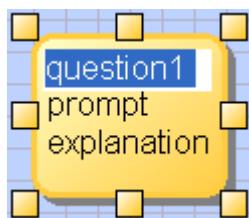
Editing Text

The text in any object within the chart can be edited simply by clicking on the text to enter edit mode.

You can drag an object in Pointer mode – but you cannot link from it.

There are several ways to finish editing an object:

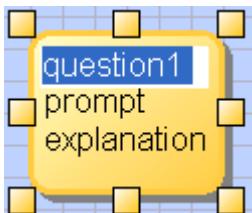
Save changes and exit edit mode; object still selected	Click the <Enter> or <Return>key or Click on one of the object's yellow border squares
Save changes, exit edit mode	Click on a blank area of the Worksheet
Lose changes and exit edit mode; object still selected	Press the <Esc> key
Save changes and switch edit mode to another object	Click on the other object
Save changes and switch to editing another text field within the same Question or Statement box	Click on another text field within the same object or Press the <Tab> key one or two times or Press the <Shift>+<Tab> keys one or two times



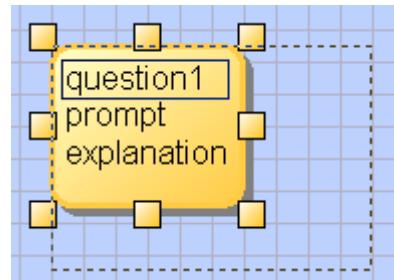
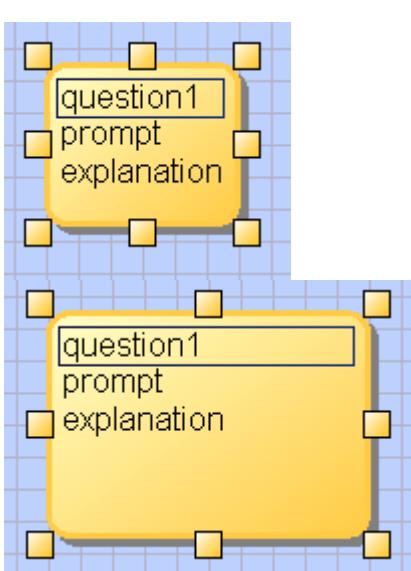
Resizing a Box?

You choose the size of a box.

The little yellow boxes allow you to drag and resize the box in any direction.



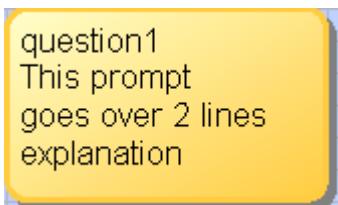
But first you have to get out of EDIT mode (for instance use the <ESC> key).



Remember, you can suppress certain fields from being displayed using the properties dialog.

But changing the size of a box, you can choose to show all or some of the text in the visible fields.

TIP: You can force new lines in text using <SHIFT> <ENTER> combination



Moving Objects

One or more selected objects can be moved simply by selecting them and then dragging the dotted outline of the objects to their new position.

Alternatively, for fine movements, you can nudge the selection by using the arrow keys. As there's a grid to snap to, this often will get rid of those ugly angled lines.

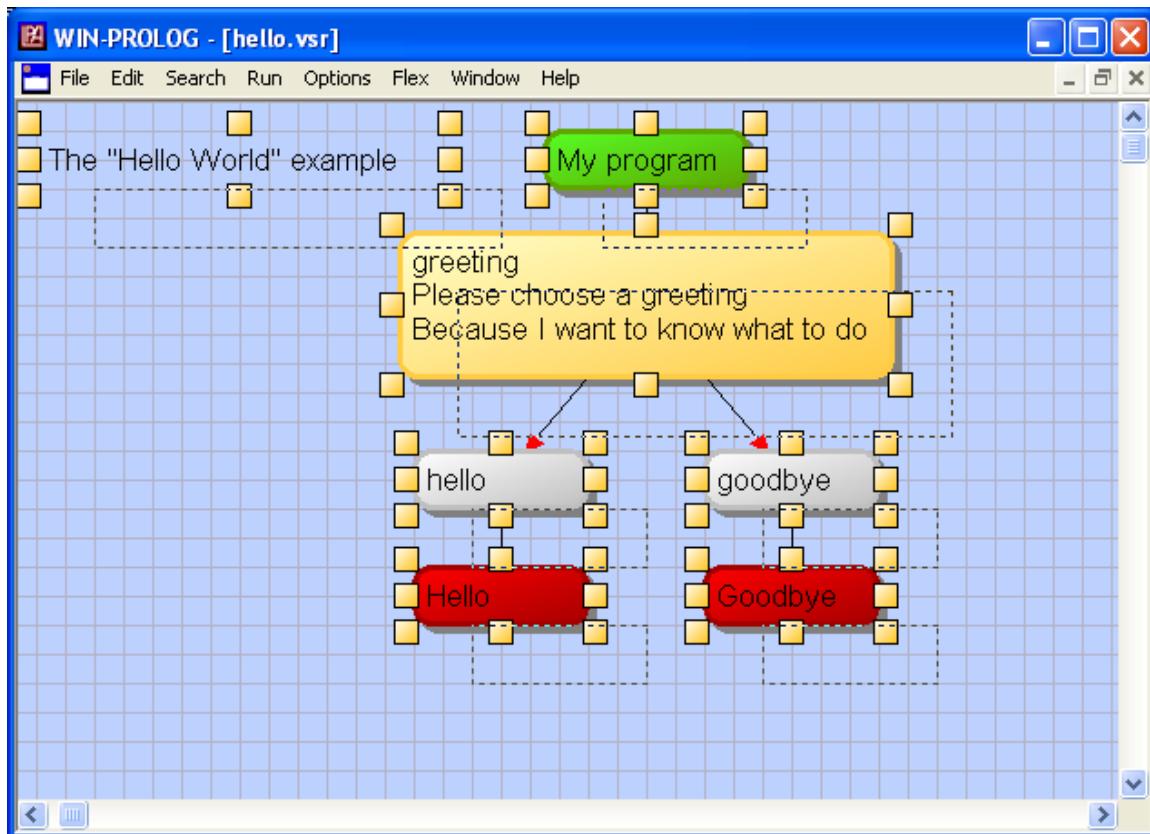


Figure 25 - Dragging multiple items

The setting of the 'Snap to Grid' option determines whether an object moves smoothly or jumps from one grid line to the next.

If the objects being moved have been vertically or horizontally aligned, such alignment should be retained following the move to a new position.

Should you move an object off the edge of the worksheet, you can select the 'Undo' option from the Worksheet menu or press **<Ctrl><A>** to select all objects and then move all the objects back into view or hold down the spacebar and pan the chart back into view.

Deleting Objects

To delete one or more objects, simply draw a marquee around them and then press the <Delete> key or click on the 'Delete' menu option.

Or select an item and delete it. VisiRule will attempt to delete any associated objects in an intelligent manner.

Linking Objects Together

The logical flow through your chart is determined by the links between the objects.

To draw a link from one box to another, go into Pointer Mode, and left-click on the first box and drag and release over the second box. You should see a link appear as you do this. As you move the mouse pointer towards the destination box, a link will appear between the first box and the mouse pointer.

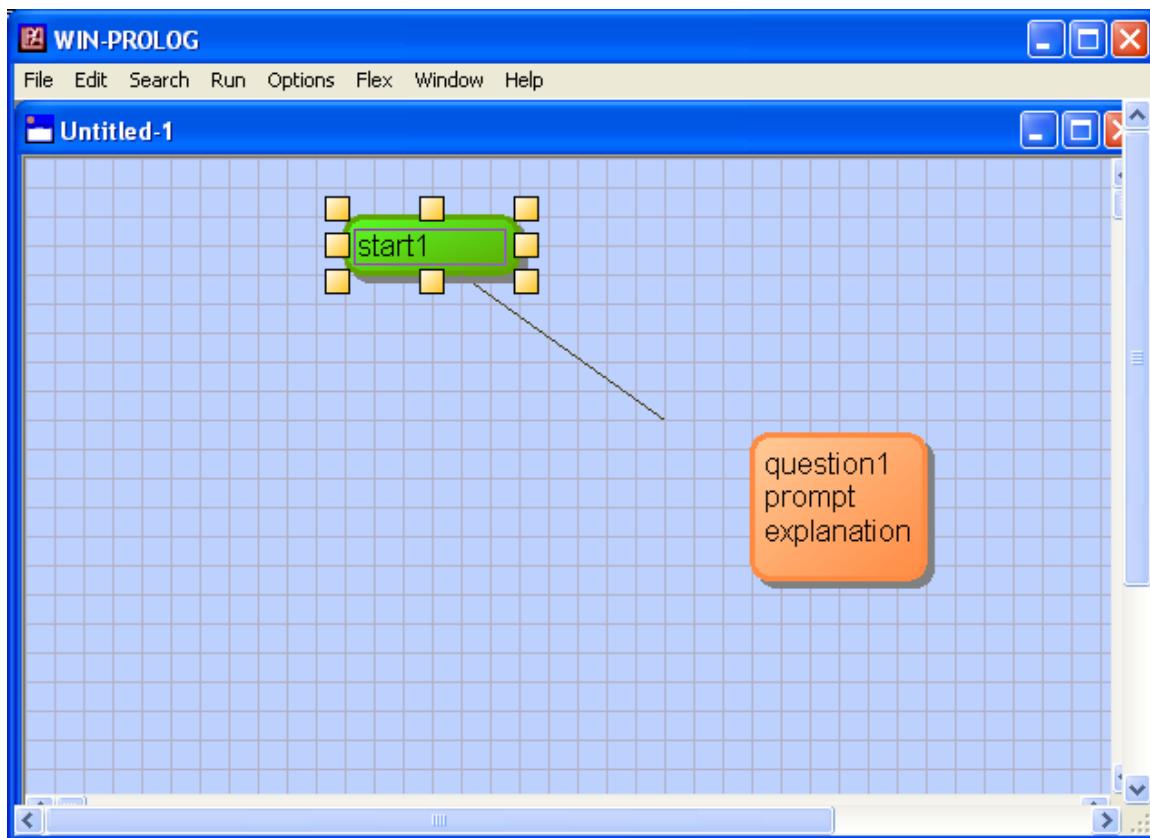


Figure 26 - Dragging a link

Move the mouse pointer over the destination box and release the button. A link will be established between the two boxes. A side effect of linking two objects is that the currently selected box switches from the first box to the destination box.

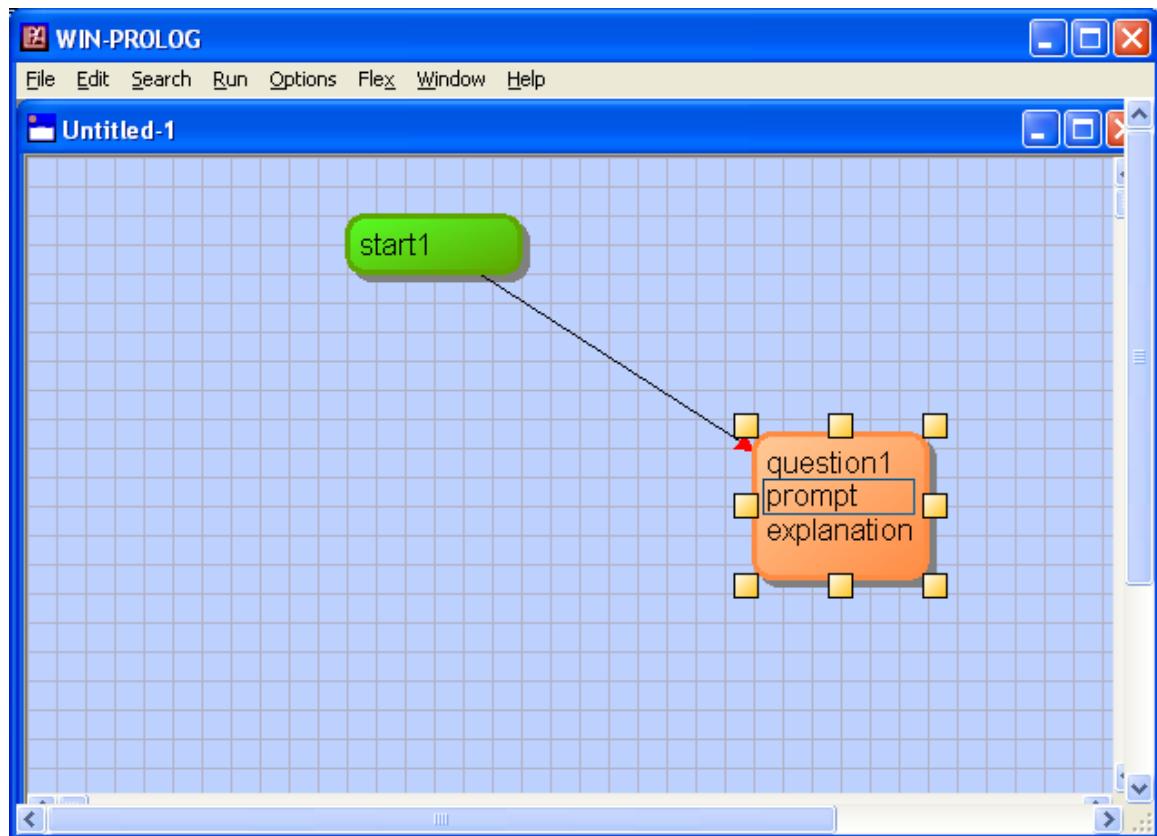


Figure 27 - Connecting a link

VisiRule will automatically try to prevent or warn you if two objects can not be linked together.



Figure 28 - Link warning message

Unlinking Objects

A link between two boxes can be deleted by right clicking on the link and selecting the 'Delete Link' option. You must not be in Pointer Mode for this to work.

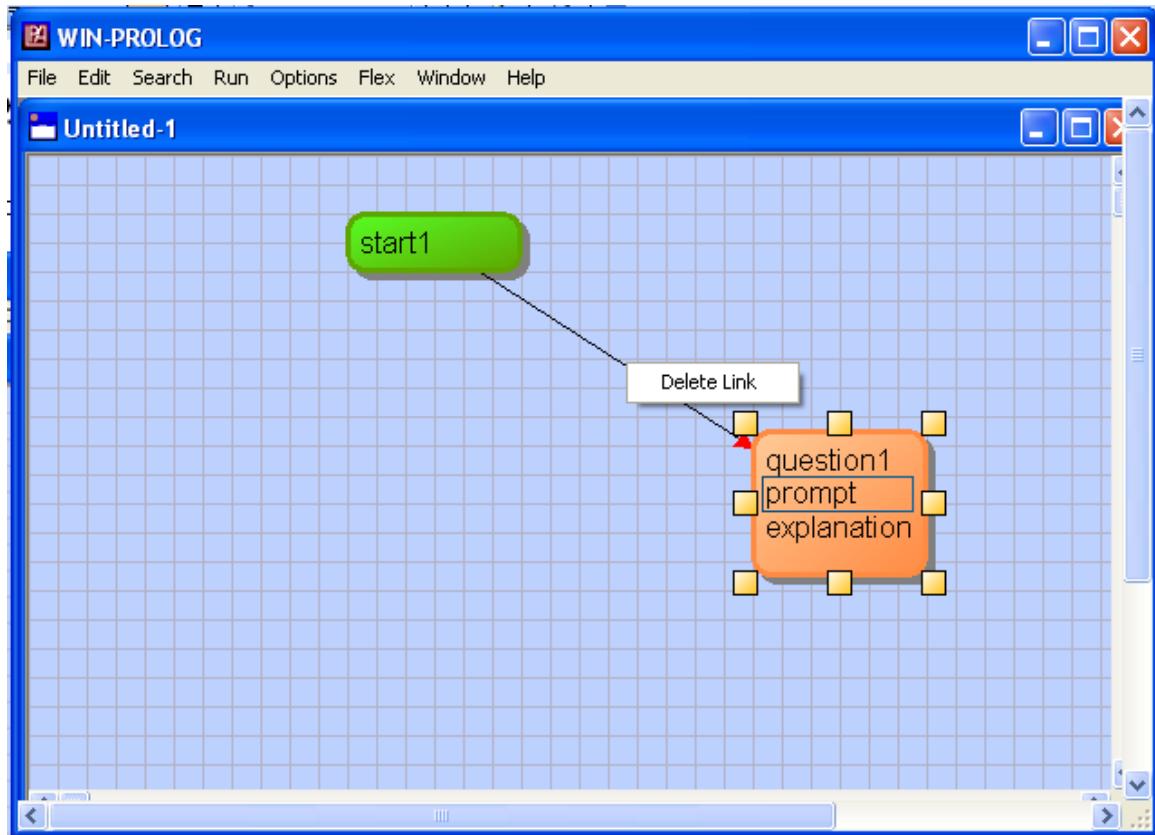


Figure 29 - About to delete a link

Smart Deletion

If you delete a box which has a link going both in and out, VisiRule will join those links back together

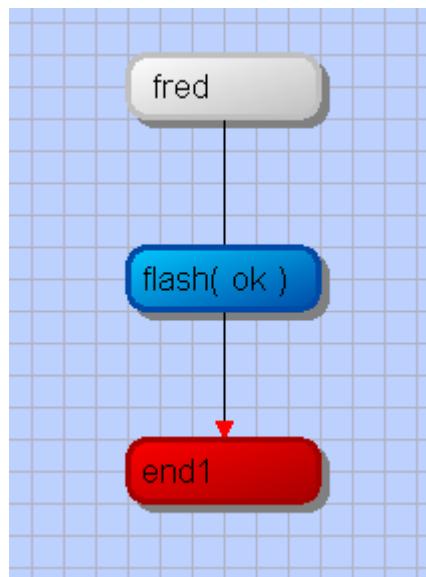


Figure 30 - Smart Deletion before

Deleting the code box will cause the links to be rejoined.

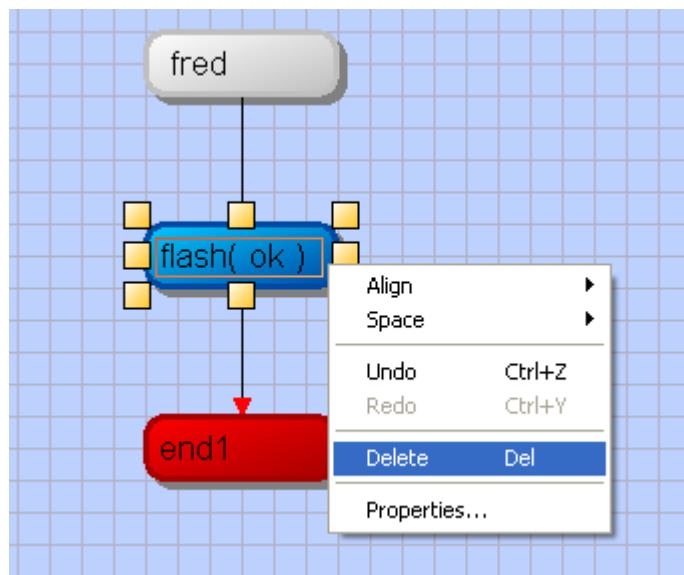


Figure 31 - About to delete

And a direct link between the two remaining boxes

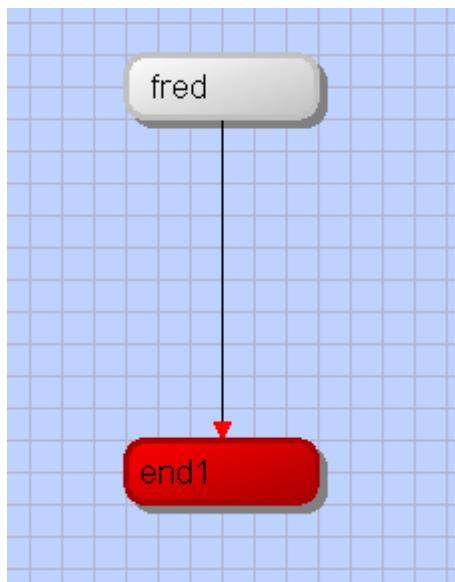


Figure 32 - Smart Deletion after

Smart Linking

You can drop a box onto a link and create two links, one going and one going out of the box we have just dropped. Say we have an expression we want to move.

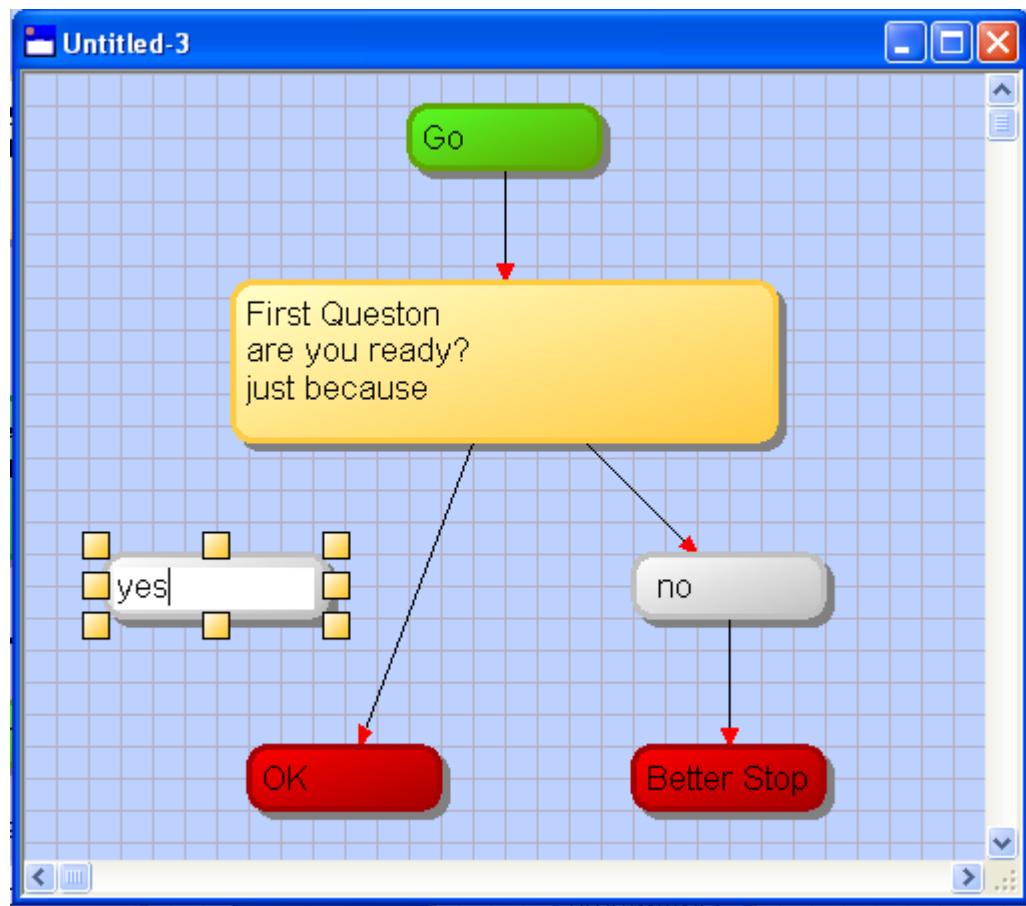


Figure 33 - Ready to drop onto link

We drag it onto the link between the two other boxes.

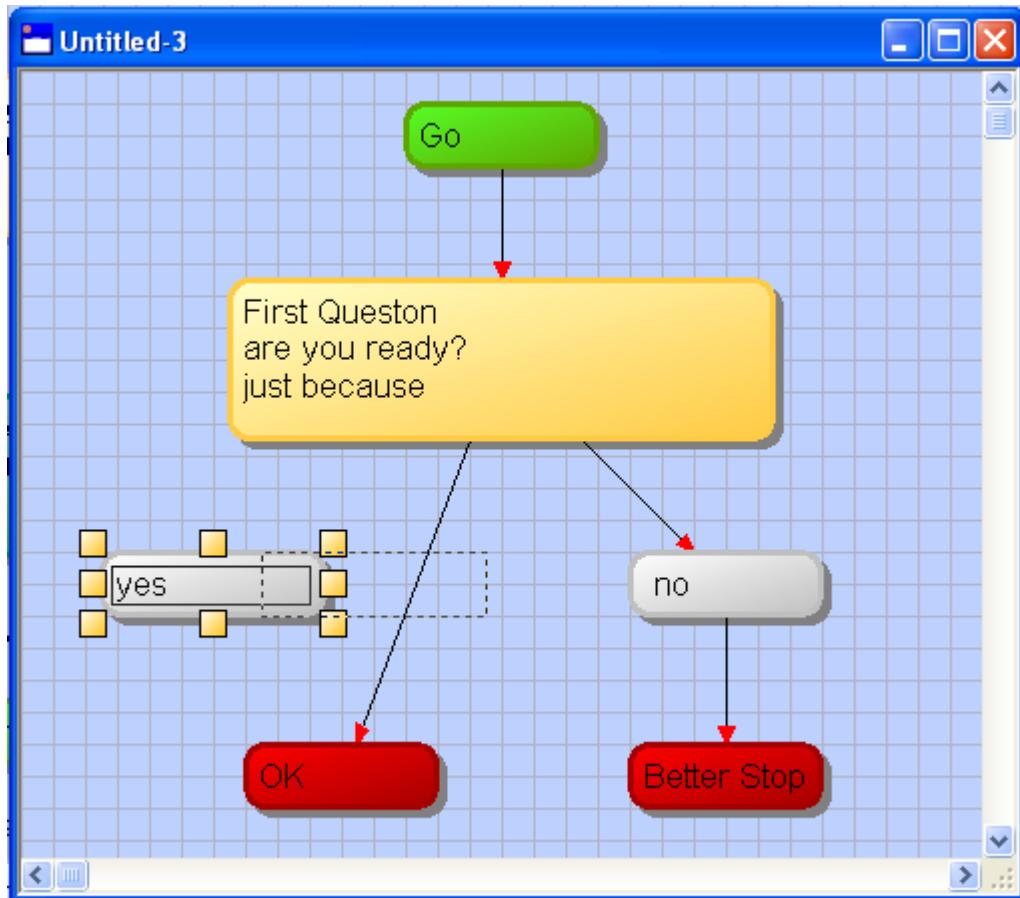


Figure 34 - Dragging the box

When we release it, VisiRule knows to make two links – one in and one out of the box we just dragged and dropped.

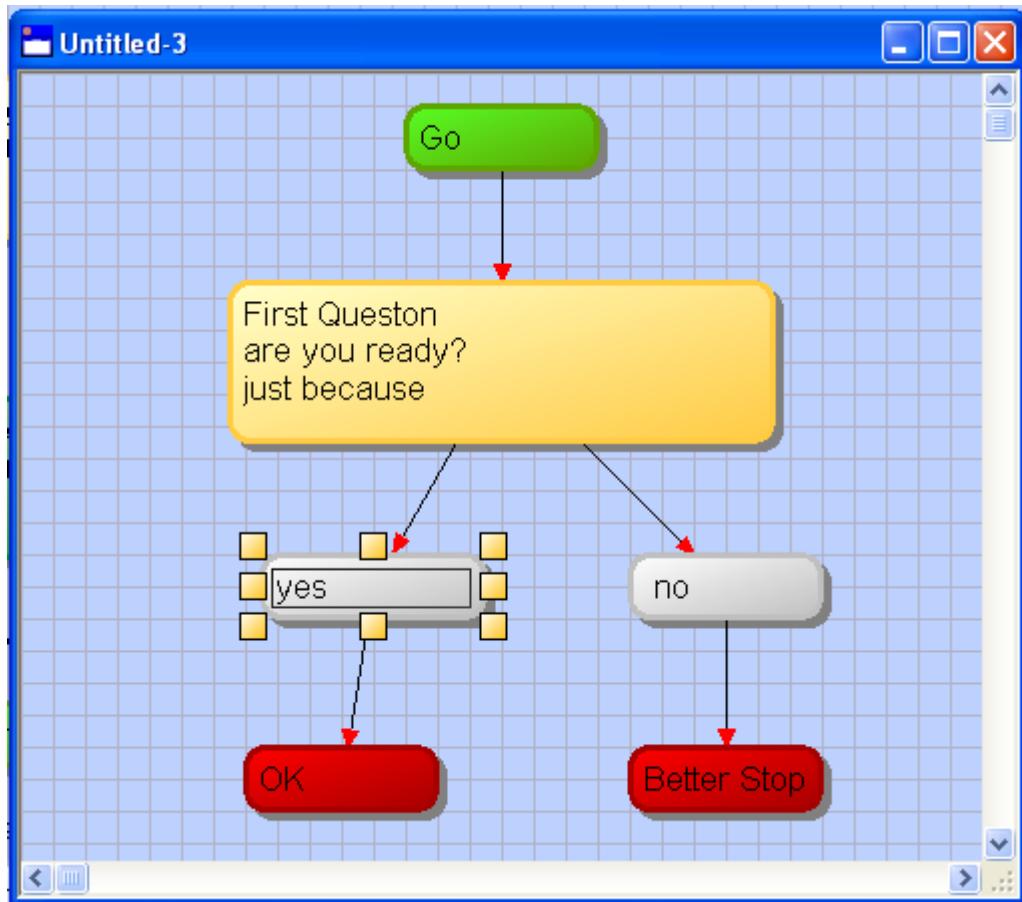


Figure 35 - Drop and split link

If the box is off-centre, just nudge it with the appropriate arrow key.

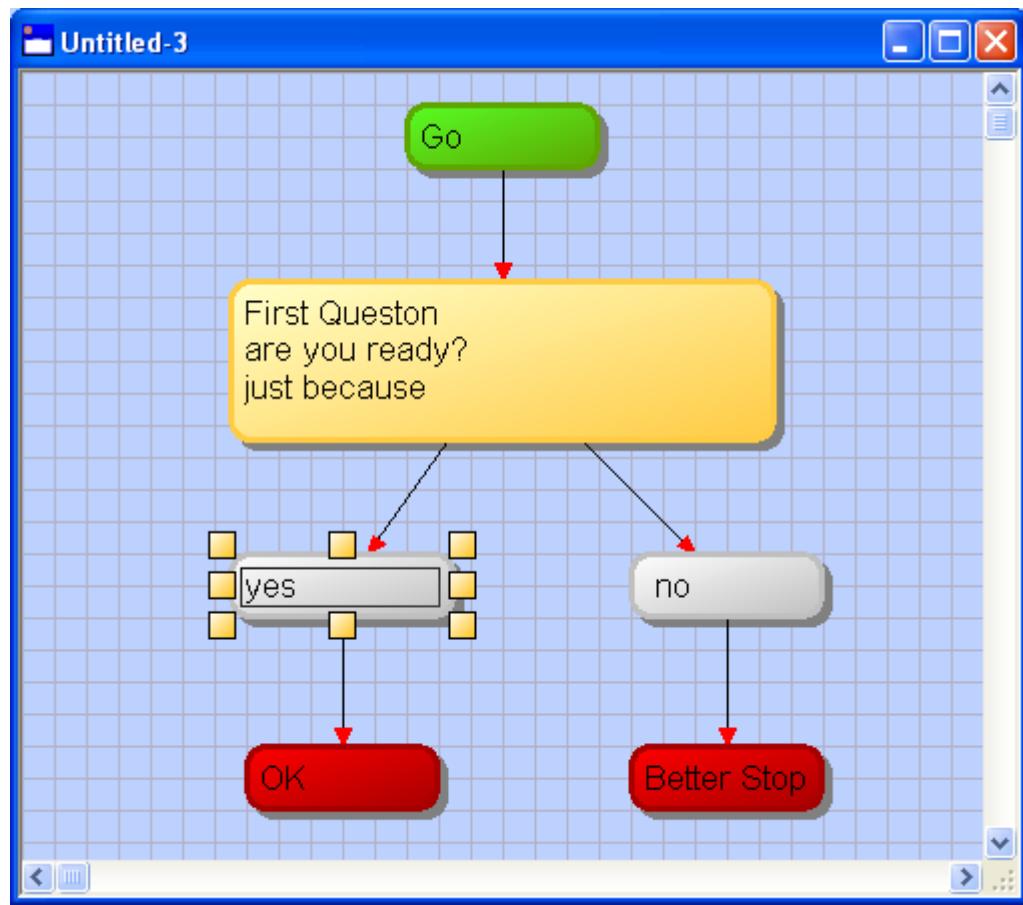


Figure 36 - Nudge box in line

More Smart Linking

Starting with three boxes

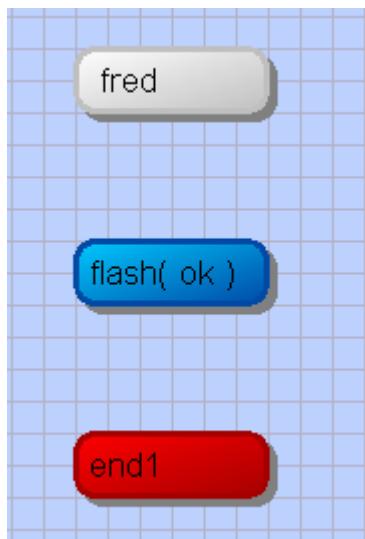


Figure 37 - Three unconnected boxes

Drag a link from the first box to the third box:

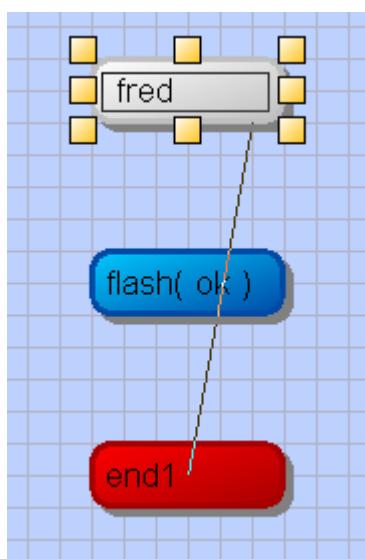


Figure 38 - Link right through them

And when released we get two links:

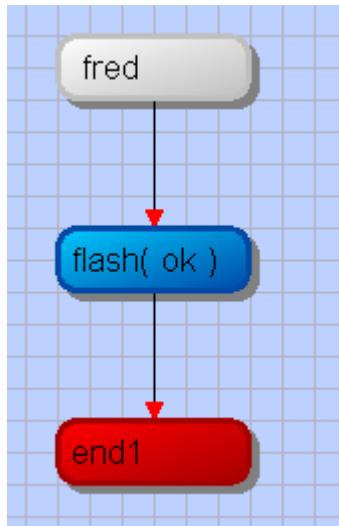


Figure 39 - After creating multiple links

Deselecting Objects

The way to deselect the currently selected objects is as follows:

- | | |
|---|---|
| Deselect all currently selected objects | Right-click on a blank area of the worksheet to display the Worksheet menu; this has the side-effect of deselecting any selected objects. |
| Remove one or more objects from a selection | Hold <Shift> down whilst dragging a marquee over the objects to be removed from the current selection. |

Navigation

To rapidly move around a large chart, hold down the <Spacebar> key; the mouse pointer will change to a hand. If you want to see what is to the right of the current display, drag the screen to the left. If you want to see what is below the current display, drag the screen upwards.

Zooming In

You can zoom into a chart by holding down <Ctrl>+<Spacebar>. The mouse pointer will change to a magnifying glass with a plus symbol in the centre. Whilst holding these two keys down, either click the left mouse button one or more times or draw a marquee over the area to be magnified.

Zooming Out

You can zoom out of a chart by holding down <Ctrl>+<Alt>+<Spacebar>. The mouse pointer will change to a magnifying glass with a minus symbol in the centre. Whilst holding these three keys down, click the left mouse button one or more times.

The Worksheet Menu

The Worksheet Menu is displayed by right-clicking on a blank area of the VisiRule worksheet.

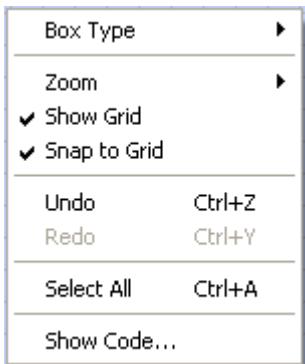


Figure 40 - Background popup menu

Displaying the Worksheet menu has two side effects – any object currently being edited will be exited with any changes saved and any object currently selected will be deselected.

The Box Type Submenu

The Box Type submenu performs the same function as the VisiRule toolbar. It contains an entry for each of the VisiRule box types with a tick against the current box type:

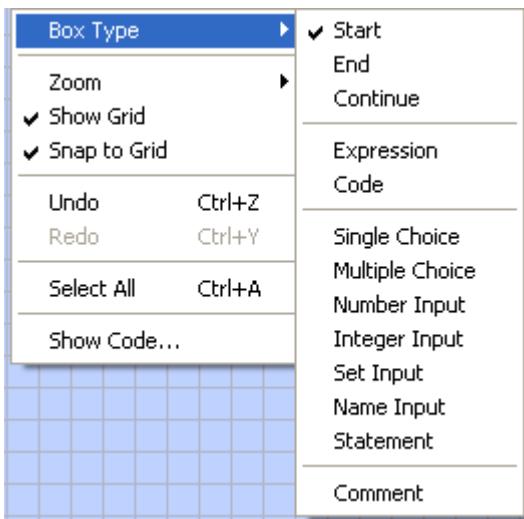


Figure 41 - Box Type popup menu

This menu is internally linked to the VisiRule Toolbar; changing the current box type on one will automatically change it on the other. You can change the current box type by clicking on any of the unticked entries.

The Zoom Submenu

The 'Zoom' option leads to the Zoom submenu. A tick will be placed against the scale currently being used on the worksheet.

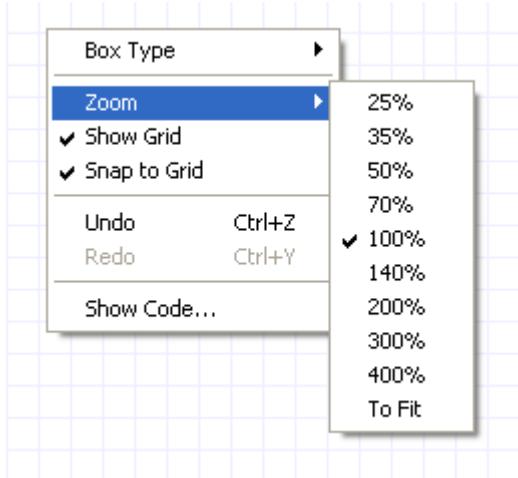


Figure 42 - Zoom popup menu

Selecting one of the other scales (except 'To Fit') will cause the worksheet to be redrawn to that scale.

Selecting the 'To Fit' option will cause the entire chart to be automatically scaled so that it fits within the Worksheet window.

Show Grid Option

The 'Show Grid' option allows you to hide or show the grid squares within the background of the worksheet. It will be ticked if the show grid option is turned on. The Snap to Grid facility is unaffected by turning the show grid option off. This option is duplicated on the Options menu.

Snap to Grid Option

The 'Snap to Grid' option, when on, forces a created, resized or moved object to snap to the grid squares. The menu entry will be ticked if the snap to grid option is turned on. This option is duplicated on the Options menu.

Undo Option

VisiRule has a multi-level undo facility. The 'Undo' option allows you to undo the creation, deletion or movement of one or more objects and/or the deletion or insertion of text in one or more edit fields. Pressing <Ctrl>+<Z> one or more times performs the same function.

Redo Option

Selecting the 'Redo' option performs one or more actions previously undone via the 'Undo' option. Pressing <Ctrl>+<Y> one or more times performs the same function.

Show Code... Option

The 'Show Code...' option allows you to compile just the chart contained within the current worksheet window. This option is duplicated on the Options menu.

The Object Menu

If you right-click over an object, the Object menu will be displayed. The menu will be slightly different depending on whether the object is currently in edit mode or not.

The Object Menu in Edit Mode

The Object Menu for editing mode is displayed by left-clicking over the text field of an object to force it into edit mode followed by a right-click. The Object menu displayed now that the object is in edit mode is as follows:

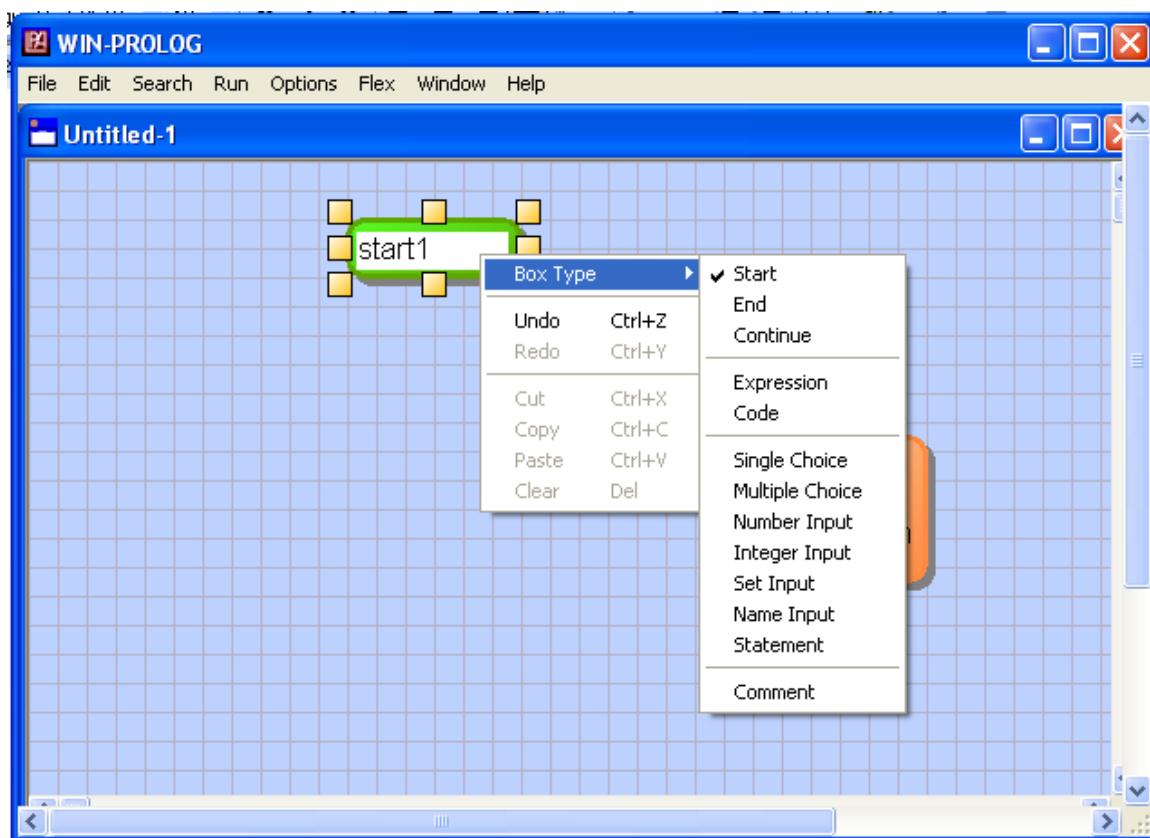


Figure 43 - About to change box type

The Box Type Submenu

The Box Type submenu contains an entry for each of the VisiRule box types:

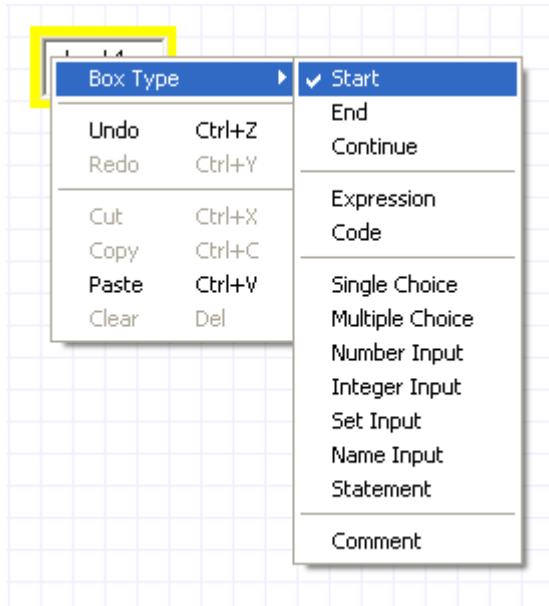


Figure 44 - Box Type Submenu

The entry for the type of box clicked upon will be ticked. You can change the type of this box by clicking on any of the unticked entries.

Undo Option

VisiRule has a multi-level undo facility. The 'Undo' option allows you to undo the deletion or insertion of text in an edit field. Pressing <Ctrl><Z> performs the same function. Sometimes, you have to escape out of an active edit field before you can Undo the last thing that you did.

Redo Option

Selecting the 'Redo' option performs one or more actions previously undone via the 'Undo' option. Pressing <Ctrl><Y> one or more times performs the same function.

Cut Option

The 'Cut' option removes the highlighted text in the field currently being edited, placing such text on the clipboard. Pressing <Ctrl><X> performs the same function.

Copy Option

The 'Copy' option copies to the clipboard any highlighted text in the field currently being edited. Pressing <Ctrl><C> performs the same function.

Paste Option

The 'Paste' option pastes the text on the clipboard at the text cursor position within the field currently being edited. Pressing <Ctrl><V> performs the same function.

Clear Option

The 'Clear' option deletes the highlighted text in the field currently being edited. Pressing the <Delete> key performs the same function.

The Object Menu in Non-Edit Mode

The Object menu for non-edit mode is displayed when you right-click over an unselected object, a single selected object or one of a group of selected objects. If the object is not already selected, it will become so. If the object clicked over is unselected, any selected objects will be deselected. The Object menu displayed when the object is not in edit mode is as follows:

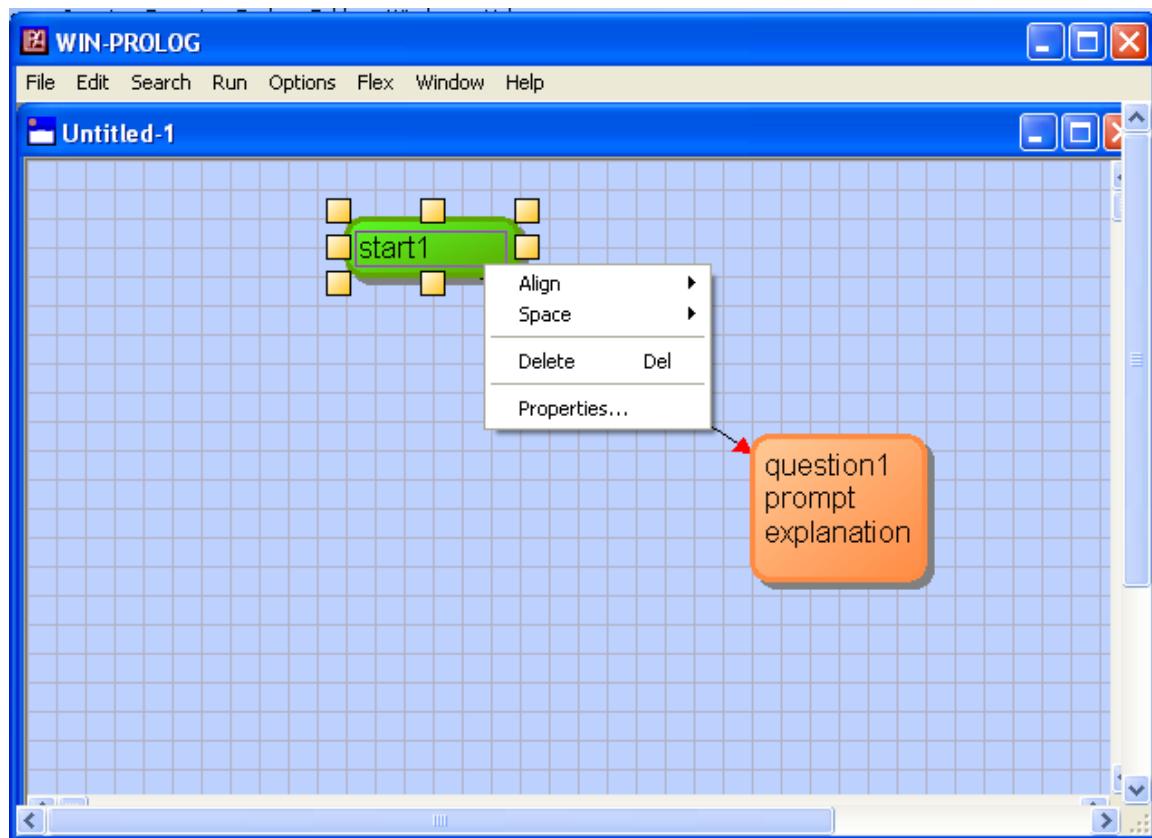


Figure 45 - Object popup menu

Tidying up your Graph using VisiRule tools

The Align Submenu

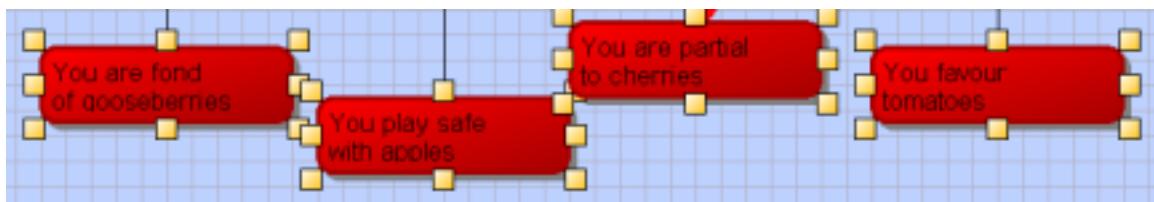
The 'Align' option leads to the Align submenu.



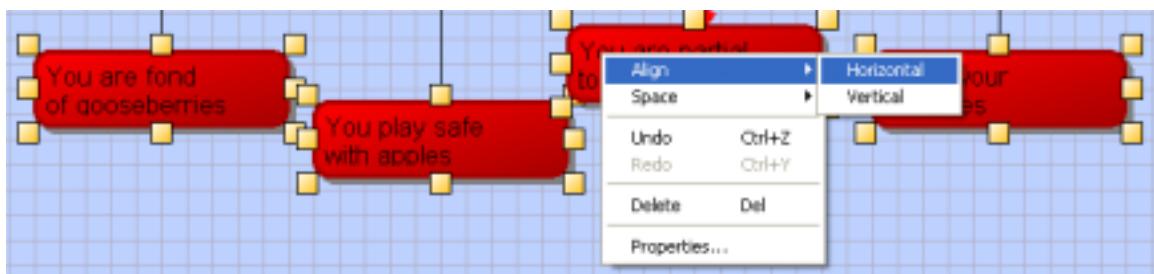
Figure 46 - Align popup menu

The 'Horizontally' menu option allows you to horizontally align two or more selected objects. For example, these objects can be changed:

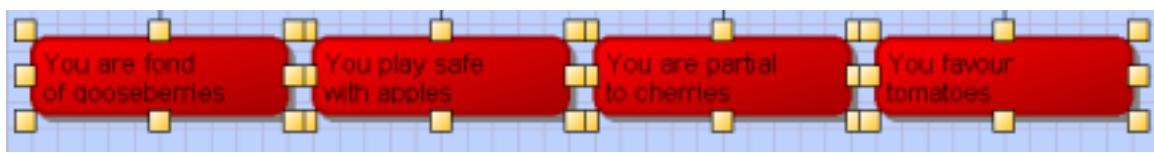
From this:



using the alignment tool



To this:



The 'Vertically' menu option allows you to vertically align two or more selected objects but leave their spacing unaltered. This function will work regardless of whether the show grid or snap to grid options are turned on or off.

The Space Submenu

The 'Space' option leads to the Space submenu.

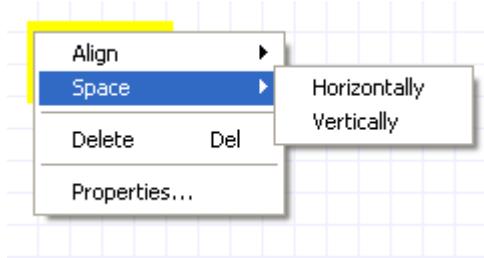


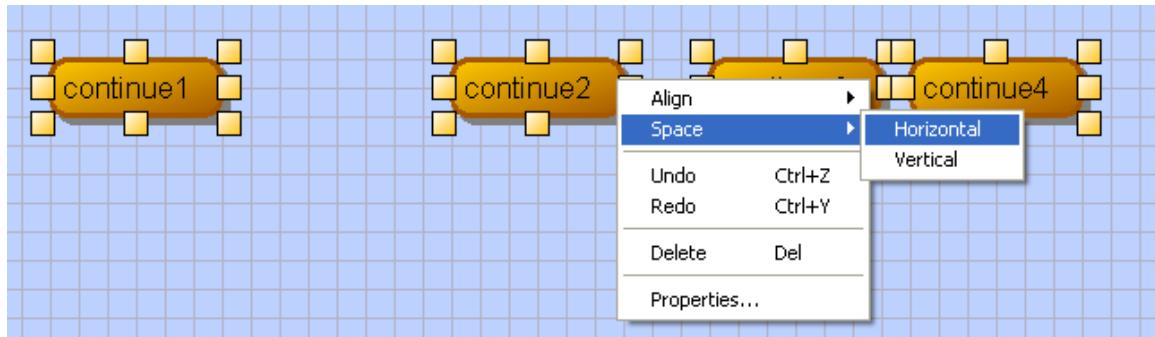
Figure 47 - Space popup menu

The 'Horizontally' menu option repositioned the two or more selected objects so that they are two grid squares apart in the horizontal direction but leaves their vertical position unaltered. For example, these four objects can be changed:

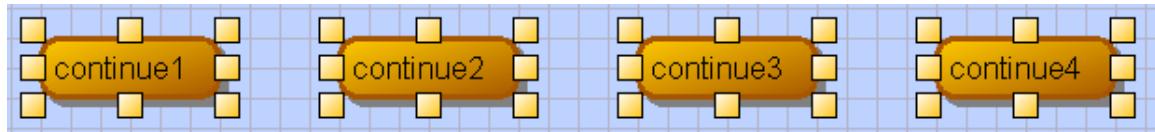
From this:



using the spacing utility:



To this:



The 'Vertically' menu option repositioned the two or more selected objects so that they are two grid squares apart in the vertical direction but leaves their horizontal position unaltered.

Delete Option

The 'Delete' option deletes all currently selected objects and any links to or from them. The DEL key is a shortcut.

Properties Option

The Properties option leads to the Properties dialog. This has two tabs - the Box Type tab and the Field Visibility tab.

In the Box Type tab, the type of the box clicked upon will be pre-selected. If a group of boxes, all of the same type, are currently selected, the single type for the group will be pre-selected. If a group of boxes of two or more different types are currently selected, no radio button will be pre-selected.

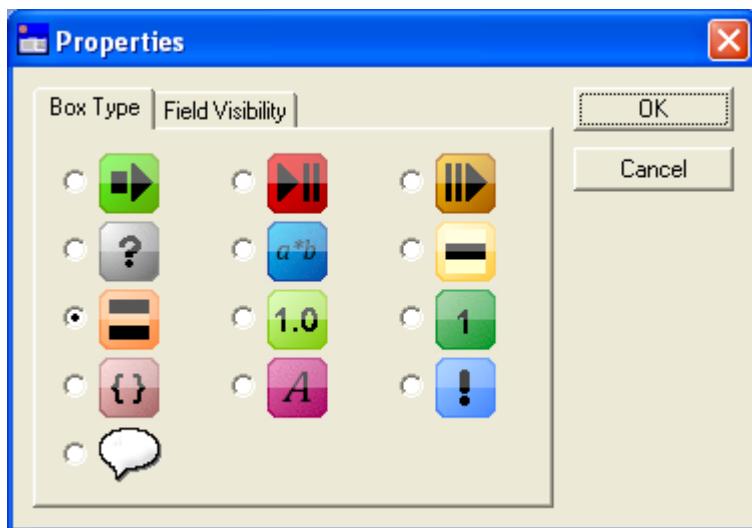


Figure 48 - Properties dialog I

If you hover over a particular icon, a tool tip will pop up telling you what box type it is.

You can change the type of one or more boxes by clicking on any of the radio buttons or icons alongside. All currently selected boxes will be changed to this new box type regardless of their original box type. When converting a box from one type to another, the text entered into it will be retained as much as possible.

The Field Visibility tab shows which fields of the objects are hidden, visible or treated in the default way. It is mainly of use in hiding the explanation field of a Question box.

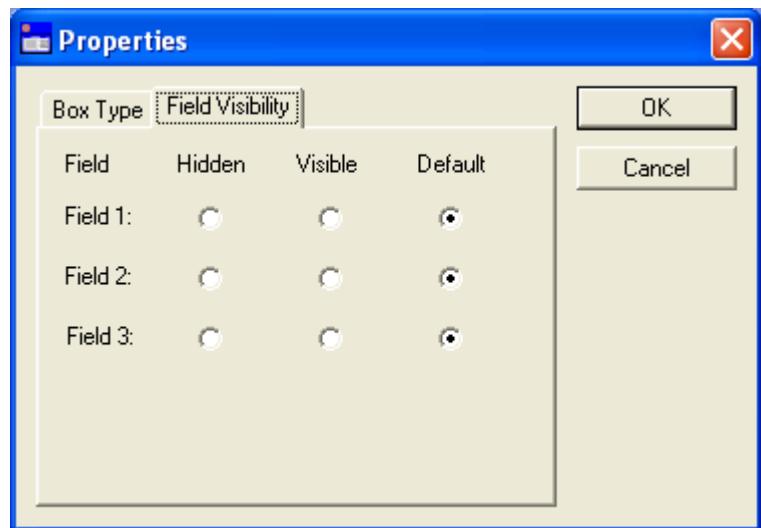


Figure 49 - Field Visibility Tab

The File Menu

New... Option

The 'New...' option allows you to open a new blank VisiRule Worksheet window. Upon clicking the 'New...' option, the New File dialog will appear:

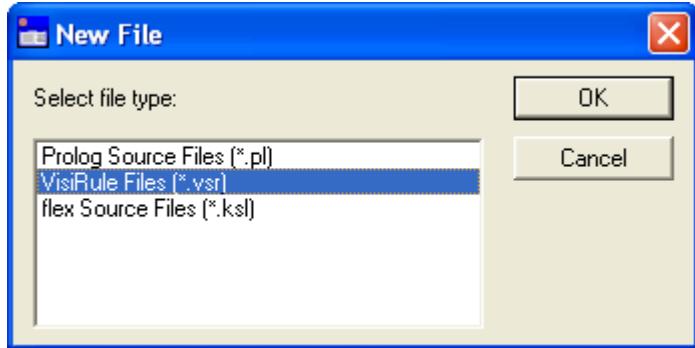


Figure 50 - File Type Dialog

Select the 'VisiRule Files (*.vsr)' entry in the listbox and click on the 'OK' button.

Open... Option

The 'Open...' option allows you to open one or more VisiRule files saved to disk. Upon clicking the 'Open...' option, the Open dialog will appear:

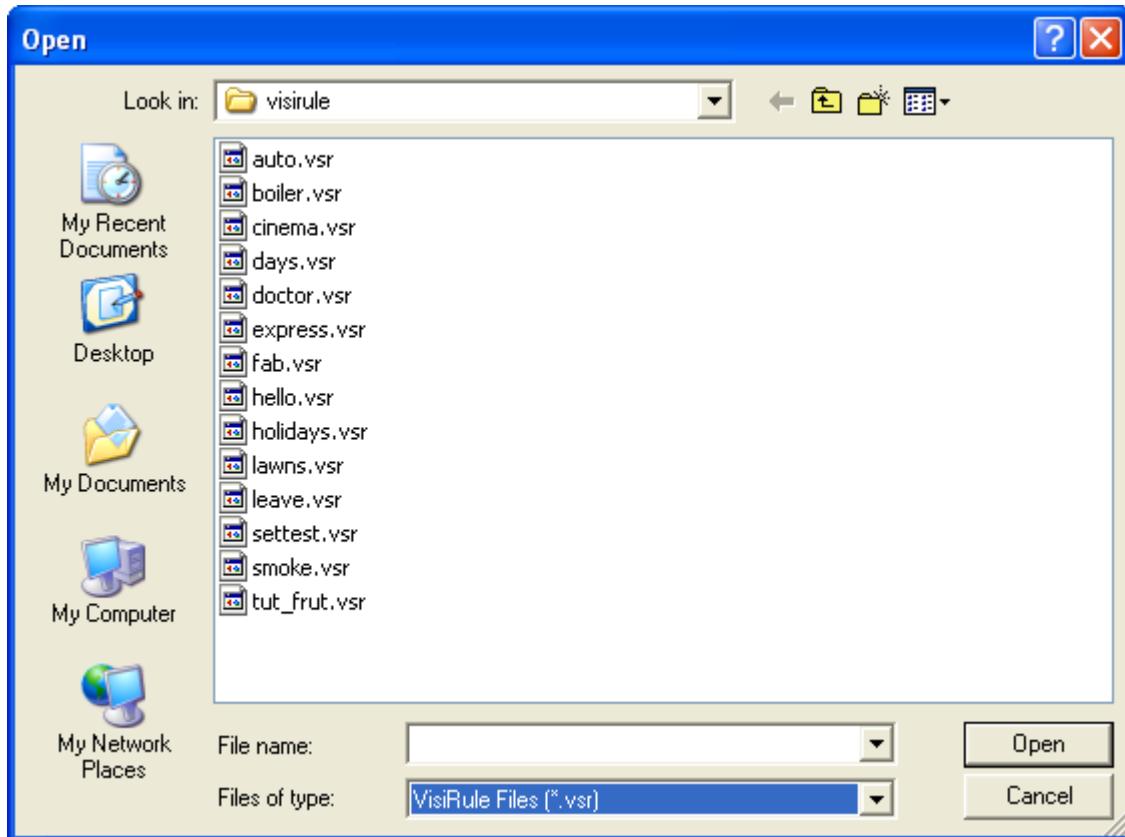


Figure 51 - Save Dialog

VisiRule files have a .VSR file extension; if required, change the 'Files of type:' field to 'VisiRule Files (*.vsr)'. Navigate to the required folder, select one or more VisiRule files and click on the 'Open' button.

Save Option

The 'Save' option allows you to save your VisiRule chart to disk. Upon clicking the 'Save' option, the Save dialog will appear:

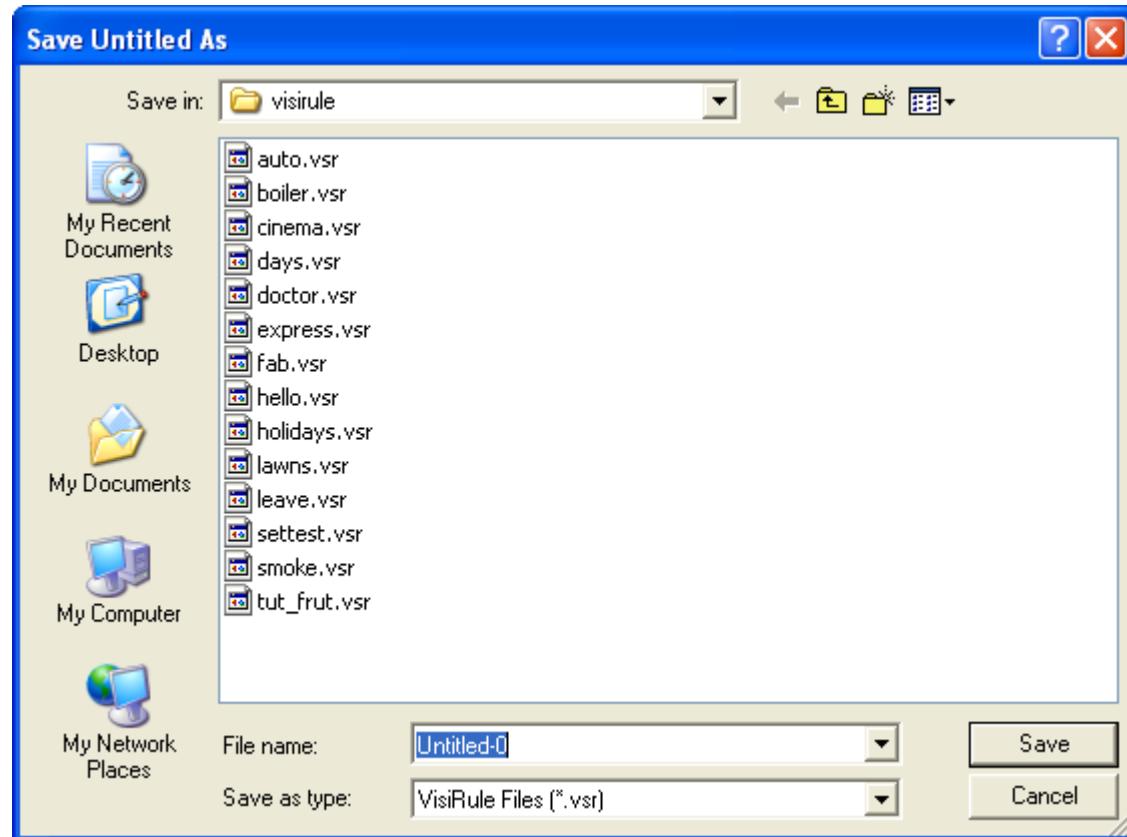


Figure 52 - Saving as VisiRule

A VisiRule file needs to have a .VSR file extension; if required, change the 'Save as type:' field to 'VisiRule Files (*.vsr)'. Navigate to the required folder, enter the name for the file in the 'File name:' field and click on the 'Save' button.

Export... Option

The 'Export...' option allows you to export your VisiRule chart as either a flex (.KSL) file or as XML or an Enhanced Windows Metafile (.WMF) file. Upon clicking the 'Export...' option, the Export VisiRule File... dialog will appear:

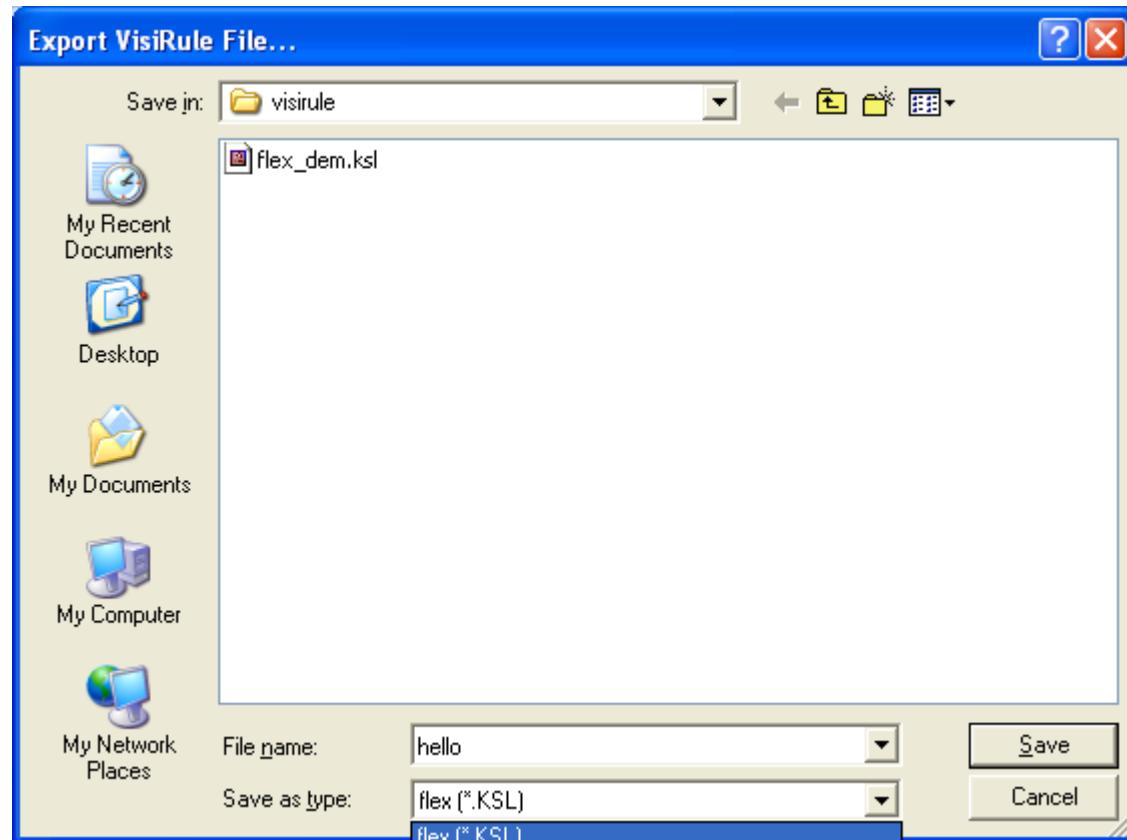


Figure 53 - Exporting as Flex KSL

Change the 'Save as type:' field to either 'flex (*.KSL)' or 'Enhanced Metafile (*.WMF)'. Navigate to the required folder, enter the name for the file in the 'File name:' field and click on the 'Save' button.

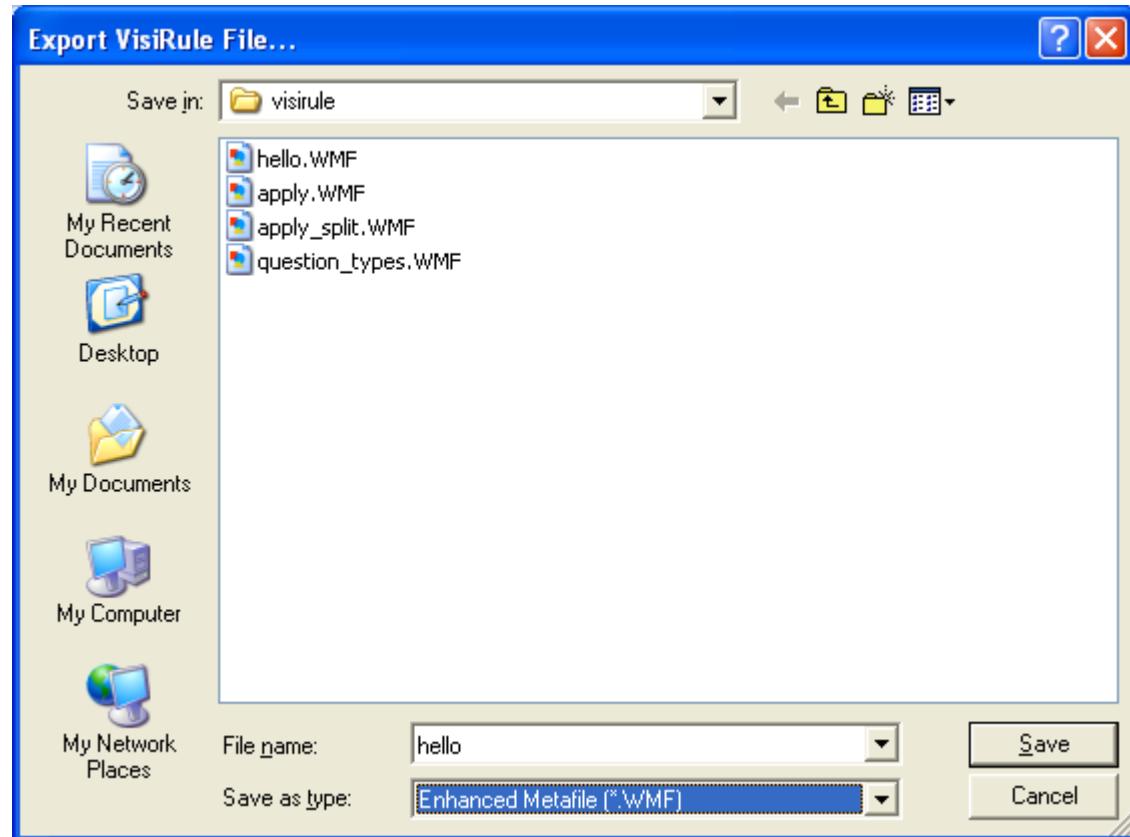


Figure 54 - Exporting as WMF

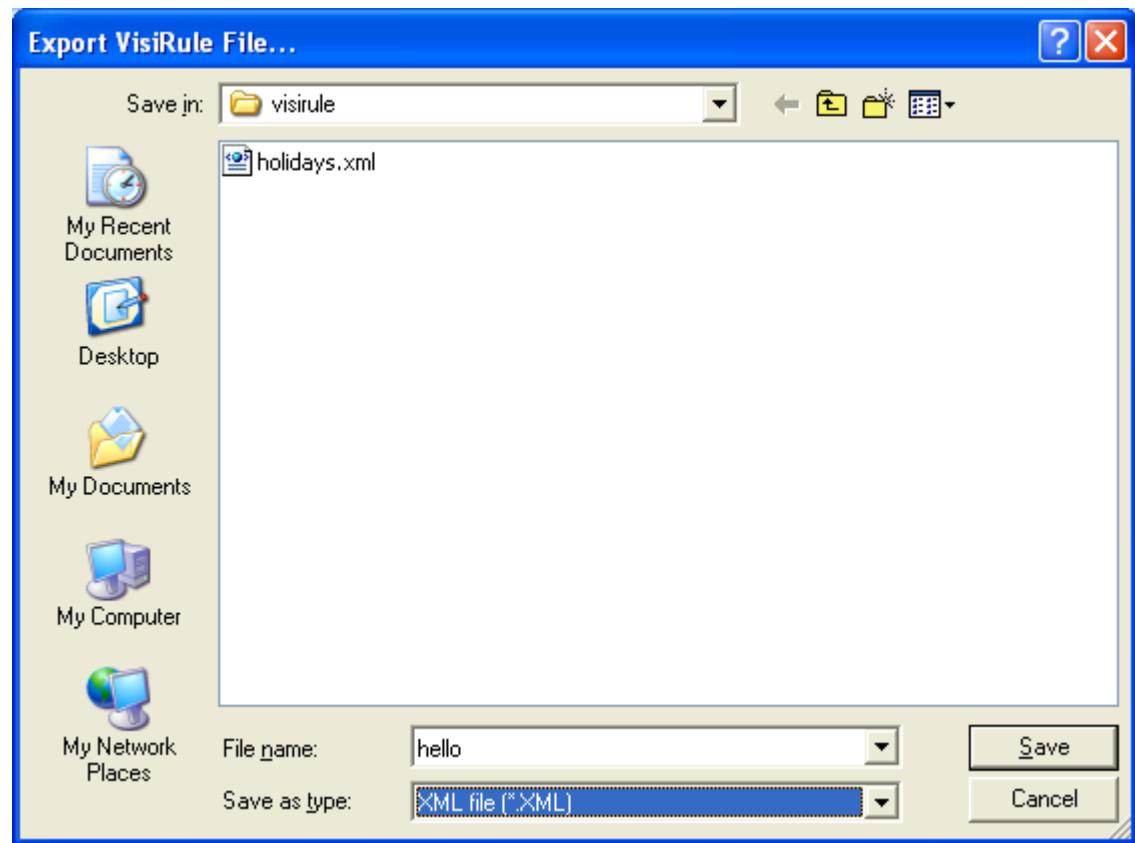


Figure 55 - Exporting as XML

The Options Menu

At the end of the Options menu are a number of VisiRule-related menu entries.

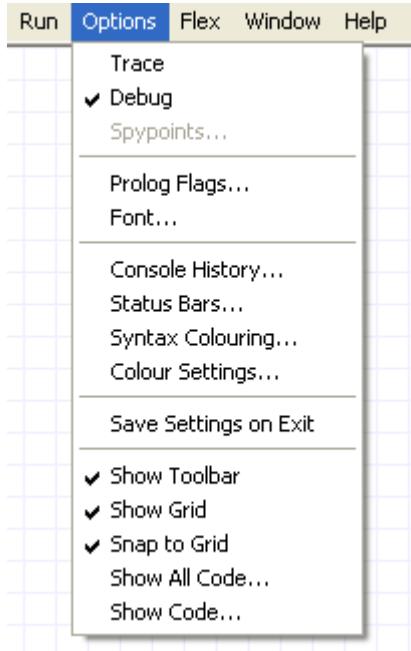


Figure 56 - Options menu

Show Toolbar Option

The 'Show Toolbar' option displays the VisiRule Toolbar. It will be ticked if the Toolbar is already displayed.

Show Grid Option

The 'Show Grid' option allows you to hide or show the grid squares within the background of the worksheet. It will be ticked if the show grid option is turned on. The Snap to Grid facility is unaffected by turning the show grid option off. This option is duplicated on the Worksheet menu.

Snap to Grid Option

The 'Snap to Grid' option, when on, forces a created, resized or moved object to snap to the grid squares. The menu entry will be ticked if the snap to grid option is turned on. This option is duplicated on the Worksheet menu.

Show All Code... Option

The 'Show All Code...' option compiles all charts currently loaded and produces a single block of code in memory. In this way you can distribute an application across multiple files using the continuation box.

Show Code... Option

The 'Show Code...' option allows you to compile just the chart contained within the current worksheet window. This option is duplicated on the Worksheet menu.

Keyboard Shortcuts

The following table shows all the VisiRule keyboard shortcuts:

Keyboard	Mode	Action
Ctrl + A	edit	Highlight all text in an edit field
	non-edit	Select all objects
Ctrl + C	edit	Copy highlighted text in an edit field to the clipboard
Ctrl + V	edit	Paste text on the clipboard at the cursor position within an edit field
Ctrl + Spacebar (hold for duration)	non-edit	Enter zoom-in mode
Ctrl + X	edit	Delete highlighted text in an edit field
Ctrl + Y	either	perform one or more actions previously undone via the <Ctrl>+<Z> undo facility
Ctrl + Z	either	Undo the creation, deletion or movement of one or more objects and/or the deletion or insertion of text in one or more edit fields
Ctrl + Alt + Spacebar (hold for duration)	non-edit	Enter zoom-out mode
Delete	edit	Delete character to right in edit field
	non-edit	Delete one or more selected objects

Escape	edit	Lose current changes when leaving edit mode
Shift + Enter	edit	Insert a new line into an edit field
Shift + Tab	edit	Toggle upwards between the three fields of a question or statement box
Spacebar (hold for duration)	non-edit	Enter navigate mode
Tab	edit	Toggle downwards between the three fields of a question or statement box

CHAPTER 5 – THE BOX TYPES

Introduction

This chapter looks at each of the VisiRule box types.

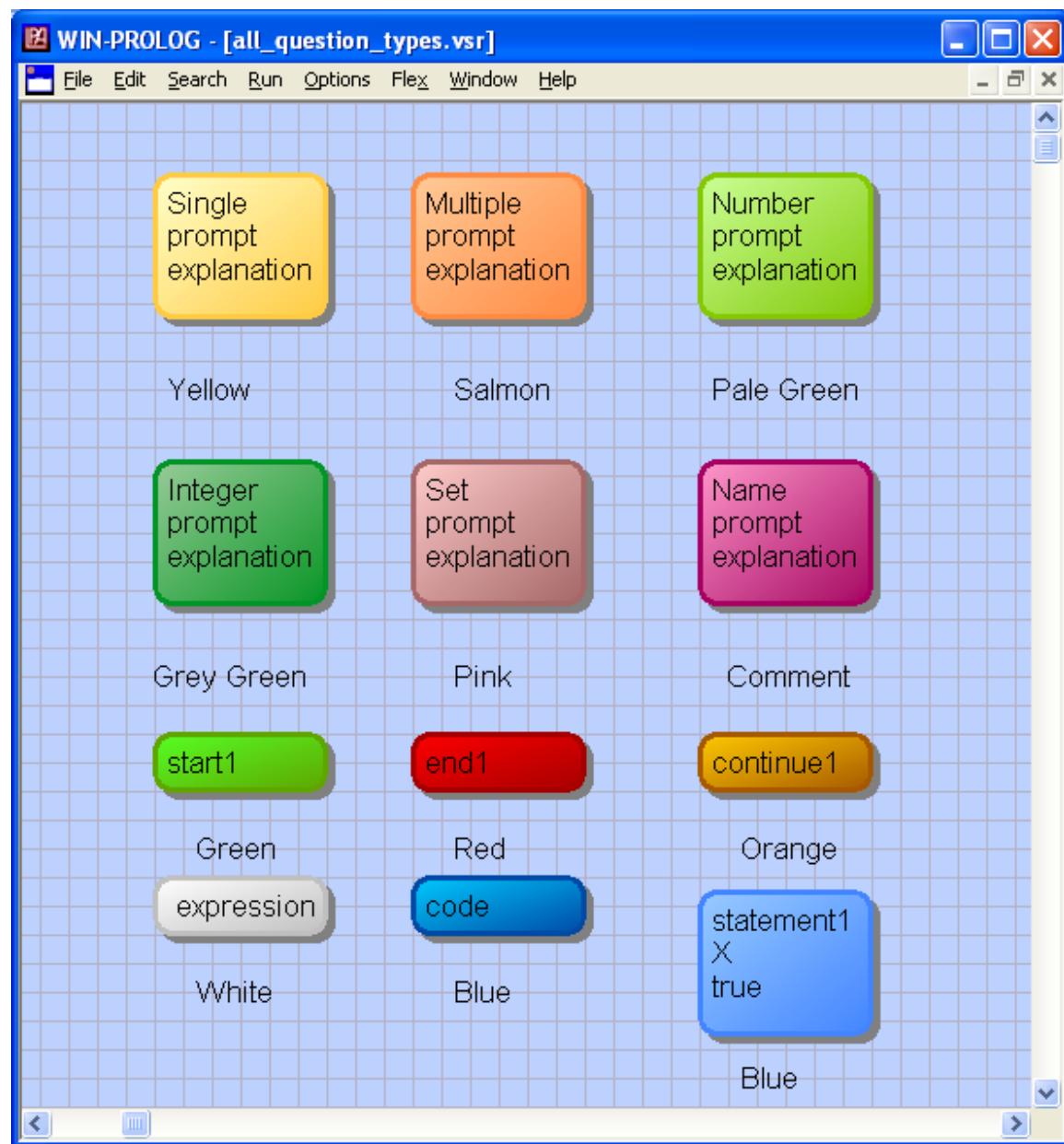


Figure 57 - VisiRule box types

Naming a Box

All the boxes in your chart must be uniquely named except for a Start box - Continue box pair. You can not, for example, have a Start box and a Question box with the same name.

Start Box

All charts begin with a Start box. However, some charts may require two or more Start boxes, each starting the chart at a different point.

The name of each Start box must be unique; by default, a Start box is given the name 'start#' where '#' is an integer. You can rename a Start box to anything you like; the name can begin with an upper case character and the convention is to use an underscore in place of a space.

The names of the Start boxes appear in the Generated flex code dialog following compilation of your chart; it is therefore sensible to give each Start box a meaningful name.



Figure 58 - Two boxes

Each Start box has one link coming out of it, going to either a Question, Statement or Code box. No box can link to a Start box.

Question Boxes

Question boxes are used to ask the user questions. They have three parameters – the a name, a prompt and an explanation.

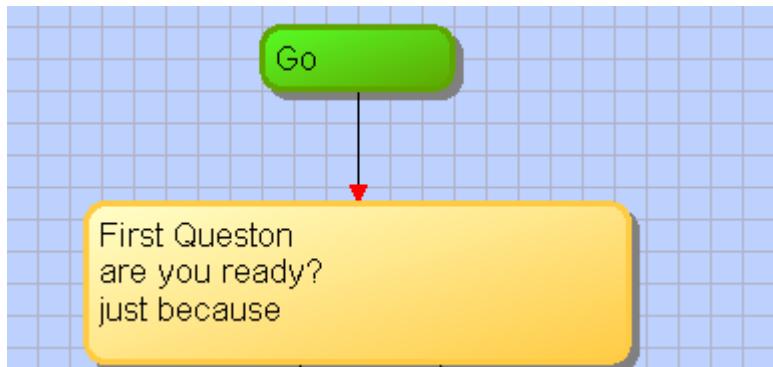


Figure 59 - Two linked boxes

The name is the question's name. It must be unique throughout the entire chart. The question name may be used later in an Expression, Statement or Code box. A space or new line character may be included in a question name; the question name will, however, need to be single-quoted within an Expression, Statement or Code box. The name is purely internal within the chart and is not seen at run-time.

Both the prompt and the explanation appear when the question is asked at runtime. These two fields may contain space and new line characters to format the text and split over two or more lines as required; use the key combination, `<Shift><Enter>`, to insert a new line at the cursor position.

The following example would produce a dialog with the prompt 'are you ready?' and an explanation button, which if pressed displays the text 'just because'.

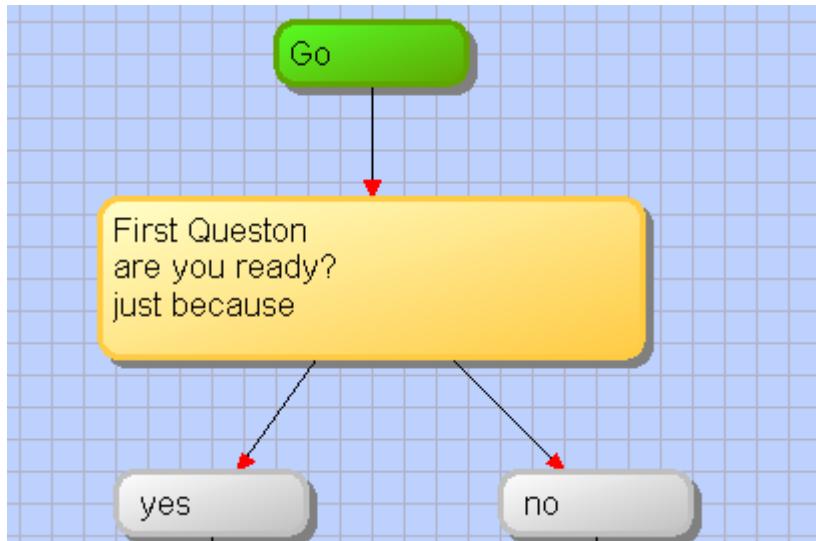


Figure 60 - Simple question

The items populating the menu in the question are automatically gathered by VisiRule in a left to right manner from expressions (e.g. **yes** and **no**) further down the same path of the chart tree. In this example, 'yes' will appear first and become the default selection.

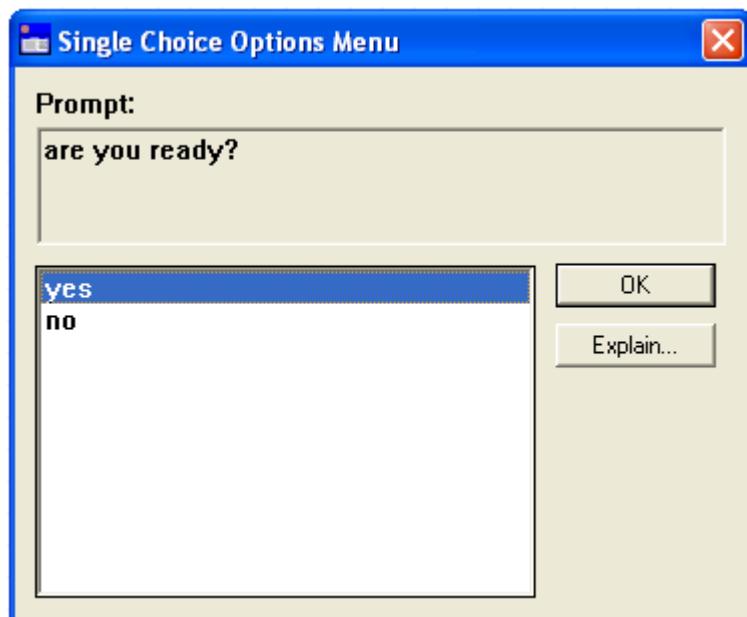


Figure 61 - Answer a question



Figure 62 - Explanation box

Question Box Types

In VisiRule, there are six types of Question box:

Single Choice The menu produced will only allow the user to select one of the items on the menu.

Multiple Choice This allows the user to select none, one or more of the items on the menu.

Number Input Instead of a menu, this option provides an input box into which the user can enter any number

Integer Input This is like Number Input, but only allows the user to enter an integer.

Set Input An input box is also provided by this option. The user can type in a list of items, each separated by a space character. For example:

red amber green

Name Input Another input box is provided into which the user can type a word or phrase.

Each of the Question box types has a different coloured background to make it easy to see which is which:

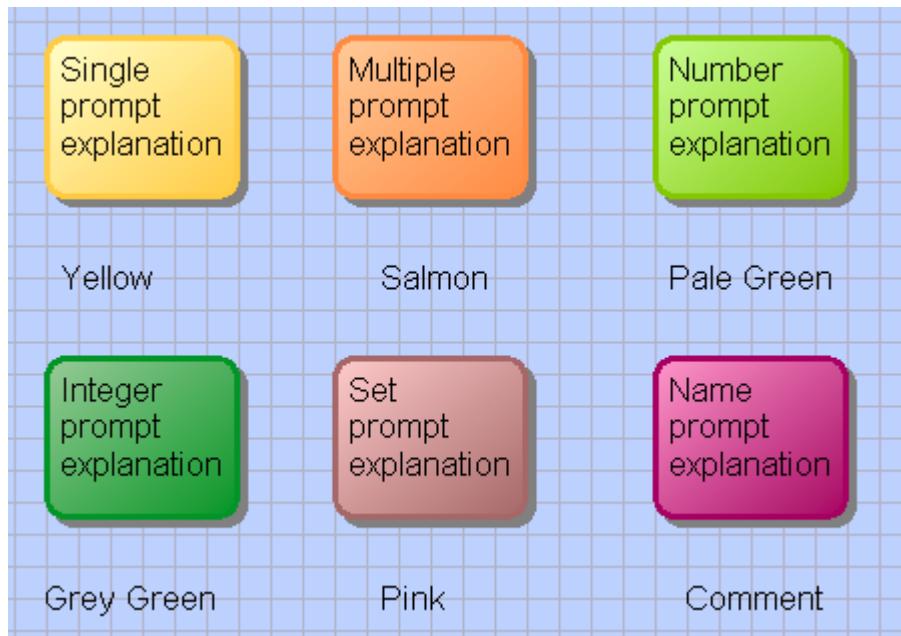


Figure 63 - Question Boxes

Not all of the elements of a Question box need to be visible. If you right-click on a Question box and select the Properties option from the menu which pops up, the Properties dialog will appear. Select the Field Visibility tab. From this tab, field 1 (question name), 2 (prompt) and/or 3 (explanation) may be hidden.

There are seven allowable combinations of hiding/showing the question name, prompt and explanation fields; it is not possible to hide all three fields at the same time.

In this way you can determine what to show at any point in time – for instance you may initially not want to clutter your chart up with explanations.

Single Choice Question Box

The Single Choice Question box will only allow the user to select one of the items on the menu. It is not possible to select none of the options.

Multiple Choice Question Box

The Multiple Choice Question box will allow the user to select none, one or more of the items on the menu.

Number Input Question Box

The Number Input Question box provides an input field into which the user can enter any number.

Integer Input Question Box

The Integer Input Question box provides an input field into which the user can enter an integer.

Set Input Question Box

The Set Input Question box provides an input field into which the user can type in a list of items, each separated by a space character or comma. Useful for when you want to probe what the user knows and do not want to give them a set of predetermined items to choose from. For example:

red amber green

Name Input Question Box

The Name Input Question box provides an input field into which the user can type a word or phrase.

Expression Box

Expression boxes test either a user's answer to a question defined in a Question box or a value that has been calculated by a Statement box. Expression boxes are very powerful and determine which paths through your chart are or are not executed.

Please note, when laying out your chart on the worksheet, that VisiRule will read your chart in a left to right manner when looking for values to populate a question with.

Please note that VisiRule automatically formats the expression within an Expression box; any hard space characters inserted into an expression will simply be ignored.

Three Components of a Simple Expression

Three components make up a simple expression:

Question Name	Comparison Operator	Value
?	=	amber

We can simply reduce this to **amber**

Question Component

The question mark (?) refers to the Question or Statement box immediately above.

It is optional and can be inferred by its absence.

The above example is testing whether the answer to the question is equal to 'amber'.

The '?' symbol is shorthand notation for the answer to the question immediately above. The definition of 'immediately above' may be over two or more levels of Expression boxes so long as no other box type appears in-between.

Sometimes, you may want to refer to a Question or Statement box defined further up the tree. A simple '?' is inadequate as VisiRule needs to know the name of the question or statement you are referring to.

In this case you must include the question or statement name in the expression, immediately after the '?', for instance:

? traffic_lights = amber

If the question or statement name contains a space character, the question or statement name will need to be quoted within the Expression box.

? 'traffic lights' = amber

Question Component
?
? traffic_lights
? 'traffic lights'

An expression box must only refer to question or statement boxes higher up on the path; a question or statement box on another path can not be used.

In the absence of any operator, a = is assumed.

Comparison Operators

VisiRule supports six comparison operators but in two contexts. They have a different interpretation depending on whether for instance we are testing a single or multiple choice question.

The meaning of a comparison operator for an atomic value is as follows:

Comparison Operator	Meaning	Example
=	equal to	? = amber
<	less than	? < 100
>	greater than	? > 10
=<	less than or equal to	? = < 10
>=	greater than or equal to	? >= 1
\=	not equal to	? \= amber

These six comparison operators will have this meaning when used with alphabetic, numeric or integer question, such as Single Choice, Integer Input and Number Input, or statement boxes. Examples:

> 0 and < 10

?q1 > 0 and ?q2 < 10

(?q1 > 0 and ?q2 > 0) or (?q1 < 10 and ?q2 < 10)

?age = 10

?color = blue

The expressions we have so far seen above apply to questions which only have one atomic answer. When we refer to Multiple Choice or Set Input questions, we need a different notation to be able to ask if the answer to the question included some value and perhaps excluded some other value.

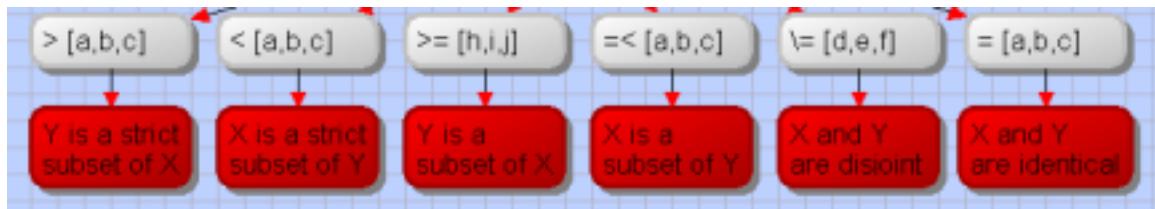


Figure 64 - Set expressions

The meaning of a comparison operator for a Multiple Choice or Set Input question is as follows:

Comparison Operator	Meaning
$=$	LHS = RHS
$<$	LHS is a strict subset of RHS
$>$	RHS is a strict subset of LHS
$=<$	LHS is a subset of RHS
\geq	RHS is a subset of LHS
\neq	LHS and RHS are disjoint – share no common elements

The following table gives the results for each comparison operator for a set question that can return one of four possible answers:

- \emptyset ,
- $[\text{walk}]$,
- $[\text{read}]$ or
- $[\text{walk}, \text{read}]$

Expression	Result of Expression Given Answer to Question
$? = \emptyset$	Only \emptyset will succeed. $[\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will fail.
$? = [\text{walk}]$	Only $[\text{walk}]$ will succeed. $\emptyset, [\text{read}]$ and $[\text{walk}, \text{read}]$ will fail.
$? = [\text{read}]$	Only $[\text{read}]$ will succeed. $\emptyset, [\text{walk}]$ and $[\text{walk}, \text{read}]$ will fail.
$? = [\text{walk}, \text{read}]$	Only $[\text{walk.read}]$ will succeed. $\emptyset, [\text{walk}]$ and $[\text{read}]$ will fail.
$? < \emptyset$	$\emptyset, [\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will all fail.
$? < [\text{walk}]$	Only \emptyset will succeed. $[\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will fail.
$? < [\text{read}]$	Only \emptyset will succeed. $[\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will fail.
$? < [\text{walk}, \text{read}]$	$\emptyset, [\text{walk}]$ and $[\text{read}]$ will succeed. Only $[\text{walk}, \text{read}]$ will fail.
$? > \emptyset$	$[\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will succeed. Only \emptyset will fail.
$? > [\text{walk}]$	Only $[\text{walk}, \text{read}]$ will succeed. $\emptyset, [\text{walk}]$ and $[\text{read}]$ will fail.
$? > [\text{read}]$	Only $[\text{walk}, \text{read}]$ will succeed. $\emptyset, [\text{walk}]$ and $[\text{read}]$ will fail.
$? > [\text{walk}, \text{read}]$	$\emptyset, [\text{walk}], [\text{read}]$ and $[\text{walk}, \text{read}]$ will all fail.
$? = < \emptyset$	Only \emptyset will succeed.

	[walk], [read] and [walk,read] will fail.
? = < [walk]	[] and [walk] will succeed. [read] and [walk,read] will fail.
? = < [read]	[] and [read] will succeed. [walk] and [walk,read] will fail.
? = < [walk,read]	[], [walk], [read] and [walk,read] will all succeed.
? >= []	[], [walk], [read] and [walk,read] will all succeed.
? >= [walk]	[walk] and [walk,read] will succeed. [], [read] will fail.
? >= [read]	[read] and [walk,read] will succeed. [], [walk] will fail.
? >= [walk,read]	Only [walk,read] will succeed. [], [walk] and [read] will fail.
? \= []	[], [walk], [read] and [walk,read] will all succeed.
? \= [walk]	[] and [read] will succeed. [walk] and [walk,read] will fail.
? \= [read]	[] and [walk] will succeed. [read] and [walk,read] will fail.
? \= [walk,read]	Only [] will succeed. [walk], [read], [walk,read] will fail.

Note: In VisiRule set notation, the comparison operators '=' and '\=' are not opposites of each other and [] \= [] is the same as [] = []; i.e. the empty set and empty share no common elements so [] \= [] succeeds as does [] = []

Now, we can also have question on the RHS and values on the left.

The Value Component

The items populating the menu generated for a Question box are automatically determined from the value component of expressions (e.g. '**yes**' and '**no**') further down the same path of the chart tree.

If a value includes a space character, it must be quoted; one convention is to use an underscore in place of a space. A value must not be enclosed in single quotes.

Combination Operators

You can combine expressions within one Expression box using combination operators. VisiRule supports two combination operators:

Combination Operator	Meaning
and	The expression on each side of the 'and' must be true. The 'and' combination operator binds stronger than 'or'.
or	Either the expression before or after the 'or' must be true.

A combination operator must be entered in lower case. A syntax error will be generated if it is entered in upper case.

Complex Expressions

The combination operators allow you to enter expressions such as '**red or green**'. Brackets allow you to override the normal precedence of 'and' and 'or'. Expressions can get quite complex as in the following example:

```
(? q1 = low or ? q1 = high) and (? q2 = cold or ? q2 = hot)
```

Layering Expression Boxes

Rather than have several complex Expression boxes, you can more boxes with less complicated expressions. This allows you to test for one thing in the first layer and, if successful, go on to the next layer.

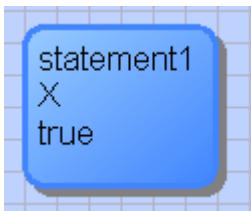
Troubleshooting

If the expression in an Expression box is shown coloured red, please check the following:

- Are you using the correct Question box type for the value you are testing for?
- Is the question or statement being tested for on the path?
- You have not entered the '>=' comparison operator as '=>'?
- You have not entered the '= <' comparison operator as '<='?
- Were the combination operators entered in lower case?
- Have you clicked into and then out of the Expression box to see if the expression turns black?
- Are the brackets correct?

Statement Box

A Statement box looks superficially like a Question box and has the same function of assigning a value to a slot. A Statement box does it by computation rather than user input; it computes a value which is then bound to the Statement box's name.



The three components of a Statement box are:

- the name. It must be unique throughout the entire chart. It can be referred to later in an Expression or Code box or another Statement box.
- an arbitrary Prolog term that will be bound to that name
- the Prolog code which instantiates any variables in the arbitrary Prolog term. This can be any flex utility or Prolog code.

In the example below, the Statement box's name, 'suit', is allocated a named suit:

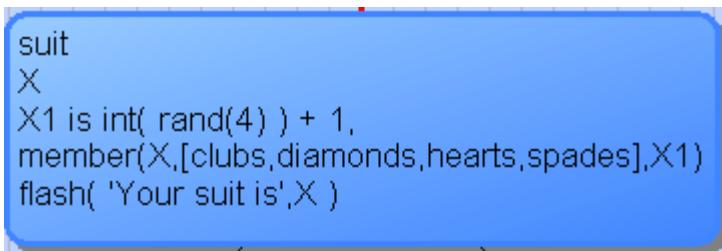


Figure 65 - Statement box for Suit

In the code part of this statement box we use:

- `is/2` and the `rand` function to generate a number between 1 and 4
- `member/3` to use that number to pick a suit from the list of suits
- `flash/N` to display the suite chosen

`X1` is used a local (internal) variable to help link the output of `is/2` and the call to `member/3`

Code Box

A Code box can be used whenever you want to include arbitrary flex utility or Prolog code in a chart.

It might be an `ensure_loaded/1` command to load a program file; or simply a call to write text out to the Console window.

A flash Command

It might be a `flash/1` command to show intermediate textual information in a dialog. A flash command in a Code box can be very useful for diagnostic purposes. It can be used to display some intermediate textual information in a dialog or prove that you have gone down a particular path in your chart.

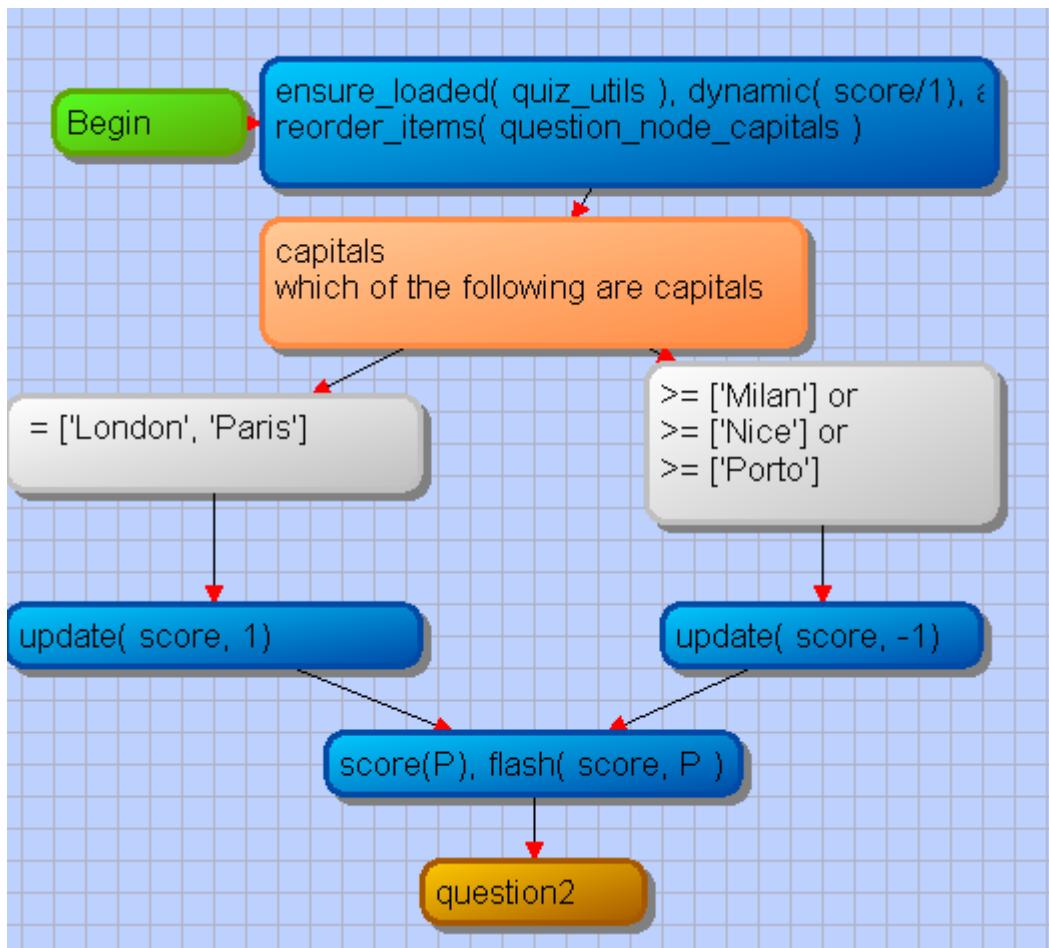


Figure 66 - Code boxes with Statement boxes

Using a Question or Statement Name in a Code Box

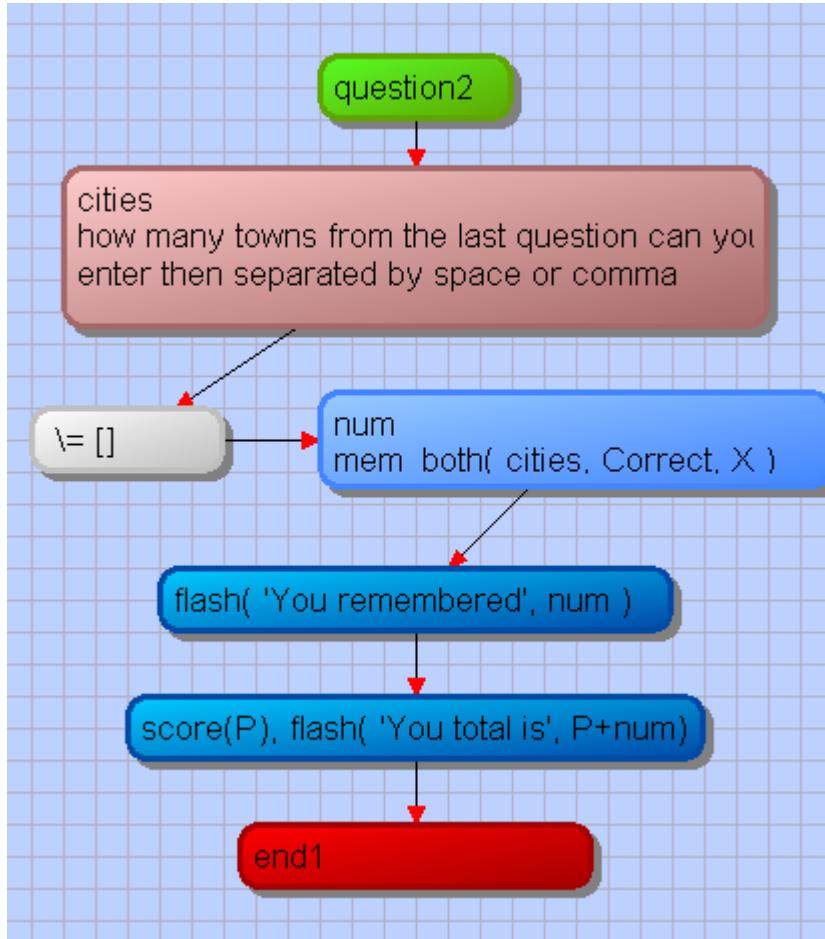


Figure 67 - Using statement names in code boxes

In the above example, we have a statement box called, num.

In the first code box, we display the value of that statement box, by:

`flash('You remembered', num)`

In the second code box, we display use value of the statement box, by:

`flash('You total is', P+num)`

Statement boxes and code boxes are extremely powerful, and coupled with the ability to include code from conventional program files, help make VisiRule such a powerful and flexible system.

Comment Box

Comment boxes are just that – a means of placing freeform text on the screen. They are very useful for adding a title to your chart or for documentation purposes. One or more space and new line characters may be inserted into a Comment box.

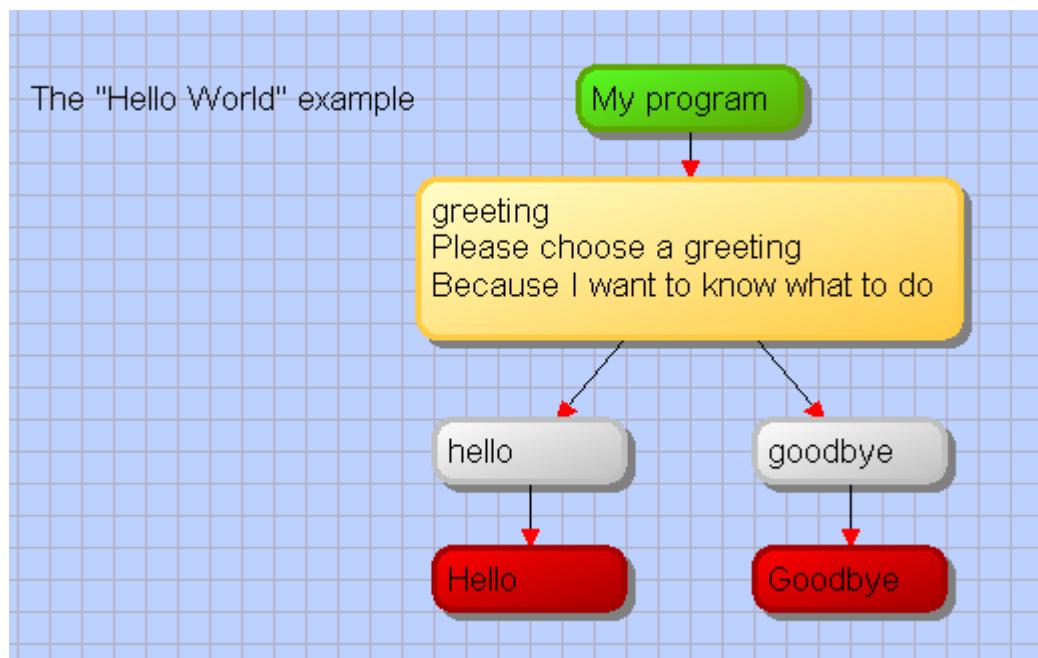


Figure 68 - Simple Comment

Continue Box

A Continue box helps to keep charts modular. The name of a Continue box must also be the name of a Start box on the same Worksheet or on another Worksheet. The following example contains two separate charts and has both a Continue box and a Start box with the name 'continue1':

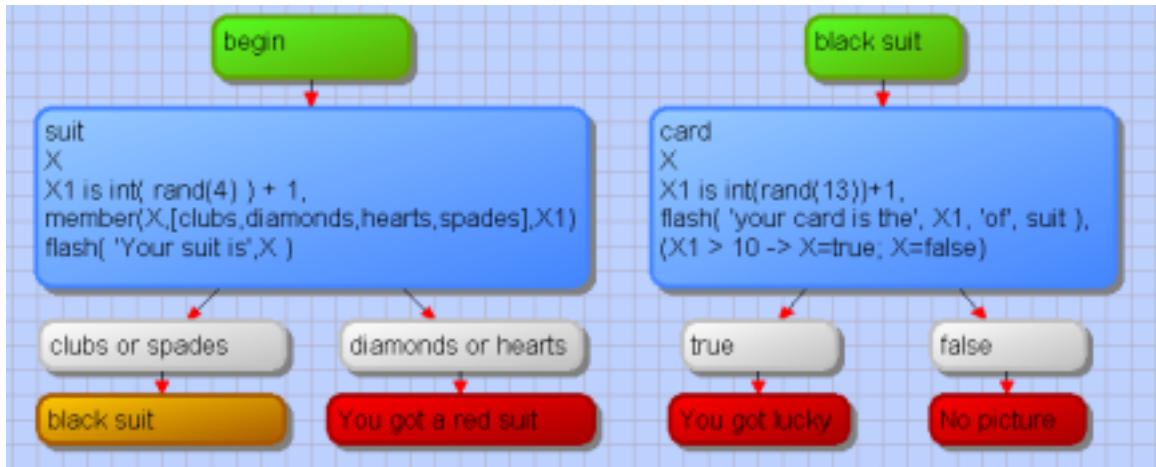


Figure 69 - Continuation Box

When execution reaches the Continue box at the end of the first chart, control switches to the top of the chart on the right.

A Continue box is in effect a pointer to some other box, a Start box of the same name. This Start box may be defined in the same file or be in a separate file. You can distribute a model across separate files using the continuation mechanism. In that way, VisiRule provides a modular development system.

End Box

An End box is the final node of a branch in the chart with the exception of branches which terminate with a Continue box. An End box, when reached, displays its name just like a flash command in a Code box.



Space and new line characters may be entered into an End box.

Two or more paths through your chart may link to (converge on) the same End box.

CHAPTER 6 – HINTS AND TIPS

When to Ask Questions

VisiRule gives you the option to ask all questions up front or ask each one as and when it is required.

The following example forces the 'water_level' question to be asked first even though its answer is not required in all branches of the decision tree:

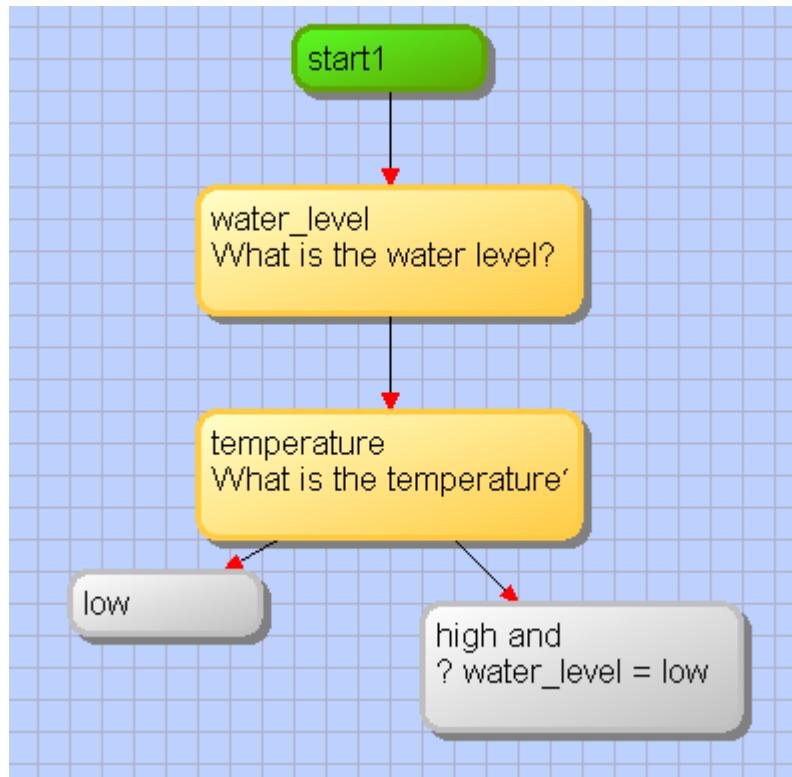


Figure 70 - Water Chart 1

It might have been more appropriate to structure the chart as follows:

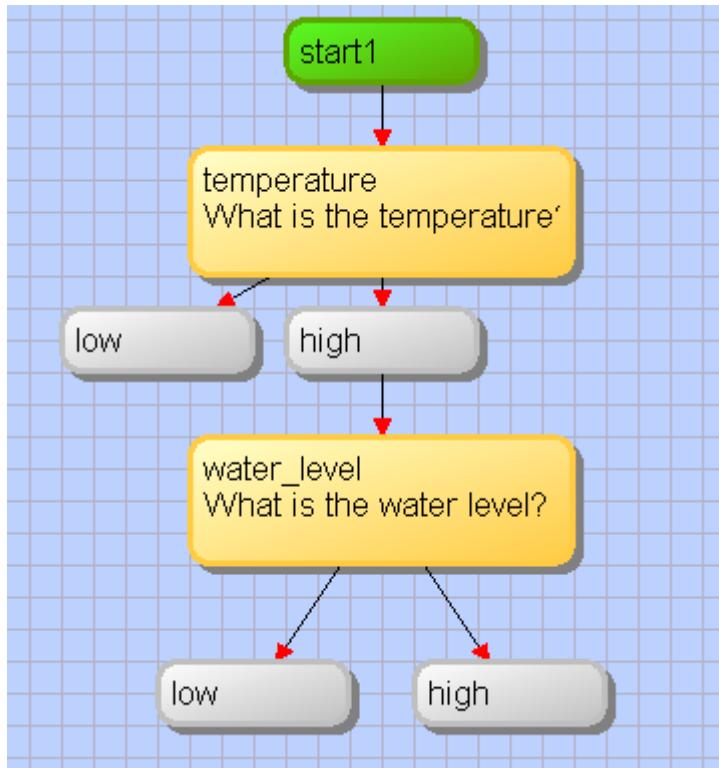


Figure 71 - Water Chart 2

In the above chart, the 'water_level' question is only asked if the temperature is high.

CHAPTER 7 – EXECUTING YOUR CHART

Compiling Your Chart

Before your chart can be executed, VisiRule has to generate some code for it and compile the code. Click on the 'Show Code...' option or the 'Show All Code...' option on the Options menu will generate flex KSL code for your chart.

The 'Show Code...' option is for when compiling the top window; the 'Show All Code...' option is for when compiling a chart split over two or more windows.

The Generated Code Dialog

If the code generation is successful, you see a dialog as such:

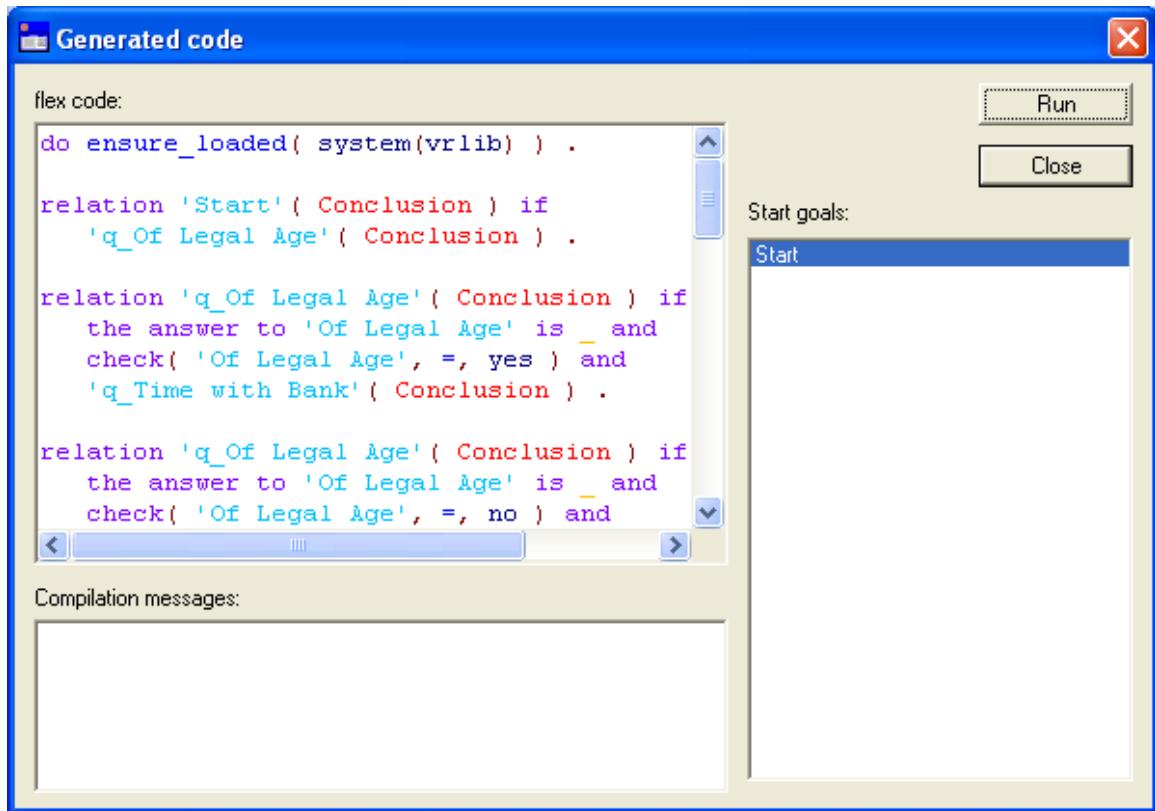


Figure 72 - Generated Code Dialog

The 'flex code:' text field on the left contains the generated flex KSL code.

Select the required Start goal from the list on the right and click on the 'Run' button. When you hit the Run button, VisiRule invokes the relevant compiler, Flex in this case, and this causes the Compilation messages field to be updated.

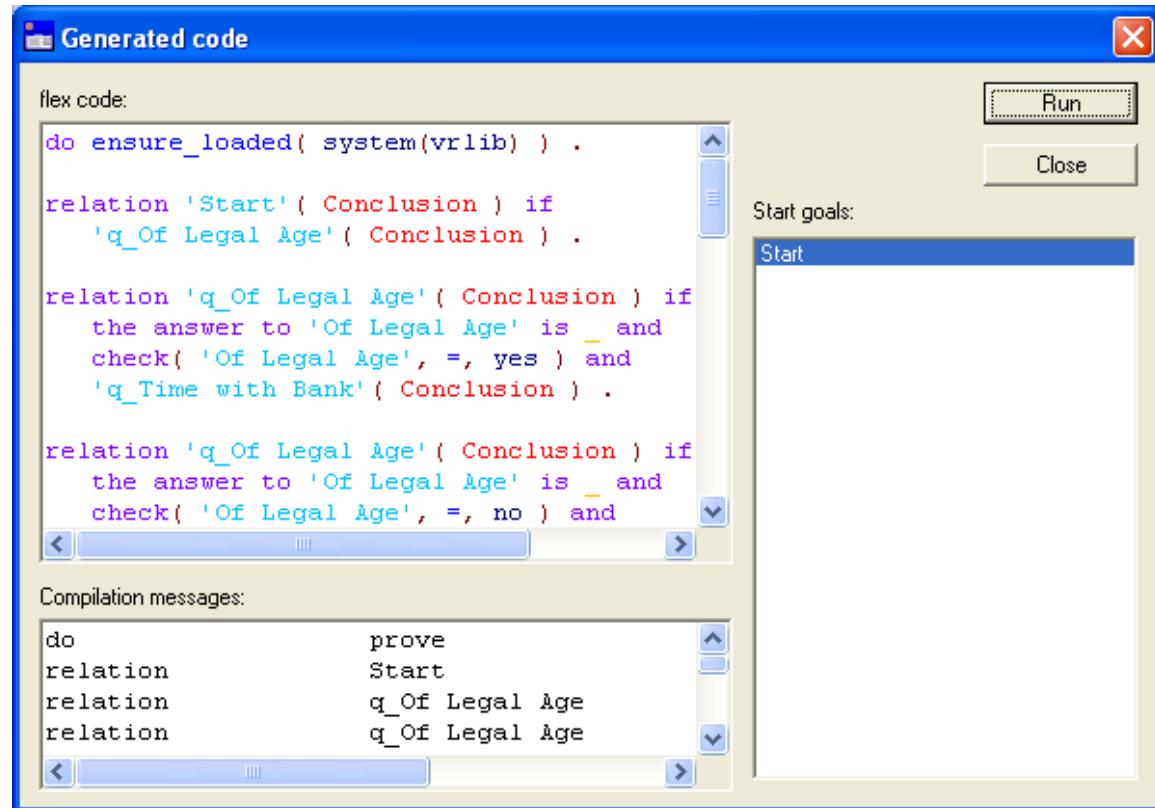


Figure 73 - Generated Code Dialog after compilation

Your VisiRule chart is now being executed; you will now be asked questions as generated from your chart's Question and Expression boxes.

The generated code may be exported in KSL format for usage within the LPA development environment. This means that, for instance, it can be combined with the Intelligence Server, to enable your chart to be embedded within a VB or Java interface or combined with WebFlex, to deliver your chart on the internet.

Running Exported KSL Code in Flex

Once the VisiRule-generated KSL code is saved to a file, it can be run independently of VisiRule, relying solely on the underlying **WIN-PROLOG** and flex.

The VisiRule-generated KSL code for the HELLO.VSR example looks as follows:

```
do ensure_loaded( system(vrlib) ) .

relation 'My program'( Conclusion ) if
    q_greeting( Conclusion ) .

relation q_greeting( Conclusion ) if
    the answer to greeting is _ and
    check( greeting, =, hello ) and
    Conclusion = 'Hello There' .

relation q_greeting( Conclusion ) if
    the answer to greeting is _ and
    check( greeting, =, bye ) and
    Conclusion = 'Goodbye' .

group group1
    hello, bye .

question greeting
    'Please choose a greeting' ;
    choose one of group1
    because Because I want to know what to do .
```

To run this particular chart, execute the command, '*My Program*/1', which corresponds to the Start box:

```
?- 'My program'( C ). <enter>
```

The question dialog will appear exactly as it does when you execute your chart from VisiRule. The variable, C, will be instantiated to the conclusion (i.e. the contents of the End box) reached.

To run your chart a second time, you will need to execute *restart*/0 first to force flex to forget the answer given to the question:

```
?- restart. <enter>
```

CHAPTER 8 – ERRORS AND DEBUGGING

Errors

VisiRule is capable of reporting errors at various stages:

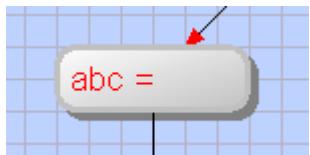
- during the editing process
- at code generation time
- at code compilation time
- at run time

Edit Time Errors

Syntax Error

An incorrect expression in an Expression box will be shown coloured red.

If you think the expression is incorrectly coloured red, because you have already corrected an error elsewhere in the chart, click in the Expression box to force into edit mode and then click elsewhere to quit edit mode and see if the expression turns black.



Code Generation Errors

If, during the code generation process, a syntax error is discovered in your chart, the offending box will be selected and scrolled to the centre of view. Unfortunately, it is not always visible, sometimes there may be something obscuring it.

Often a dialog similar to the following will appear:

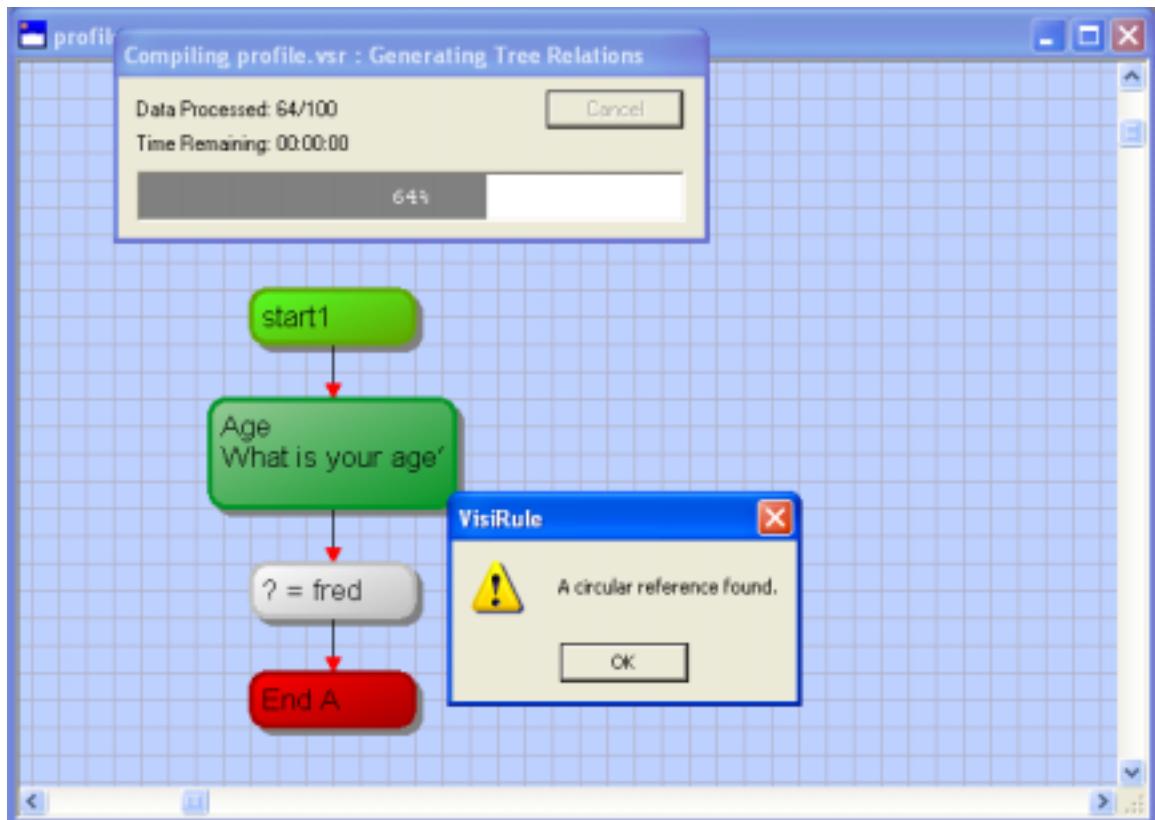


Figure 74 - Circular reference error

In the above example, it tells you that there is a problem in the Expression box containing the expression '? = fred'. The expression itself is perfectly valid but conflicts with the Question box to which it relates which requires an integer value.

Incomplete Chart

A dialog similar to the following may appear if your chart is incomplete:

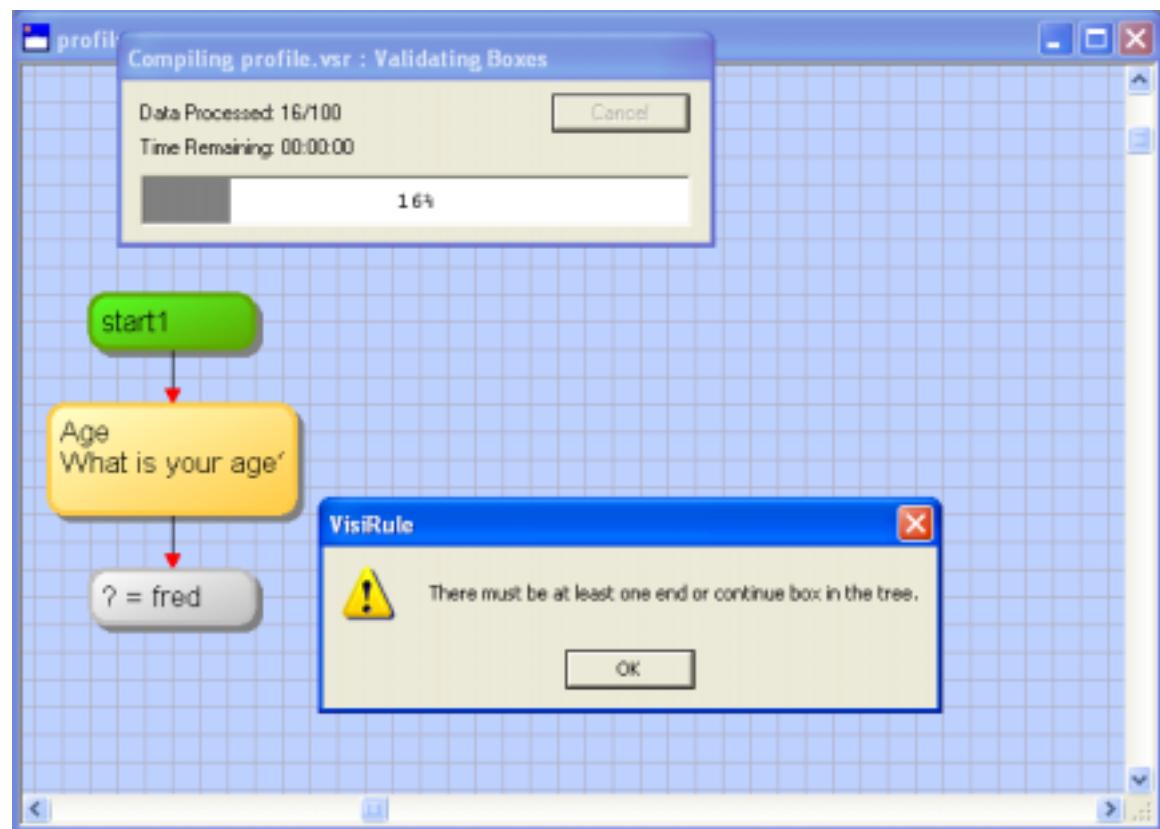


Figure 75 - Missing end box error

Syntax error in code box

A dialog similar to the following may appear if a code box is deemed to be incorrectly coded

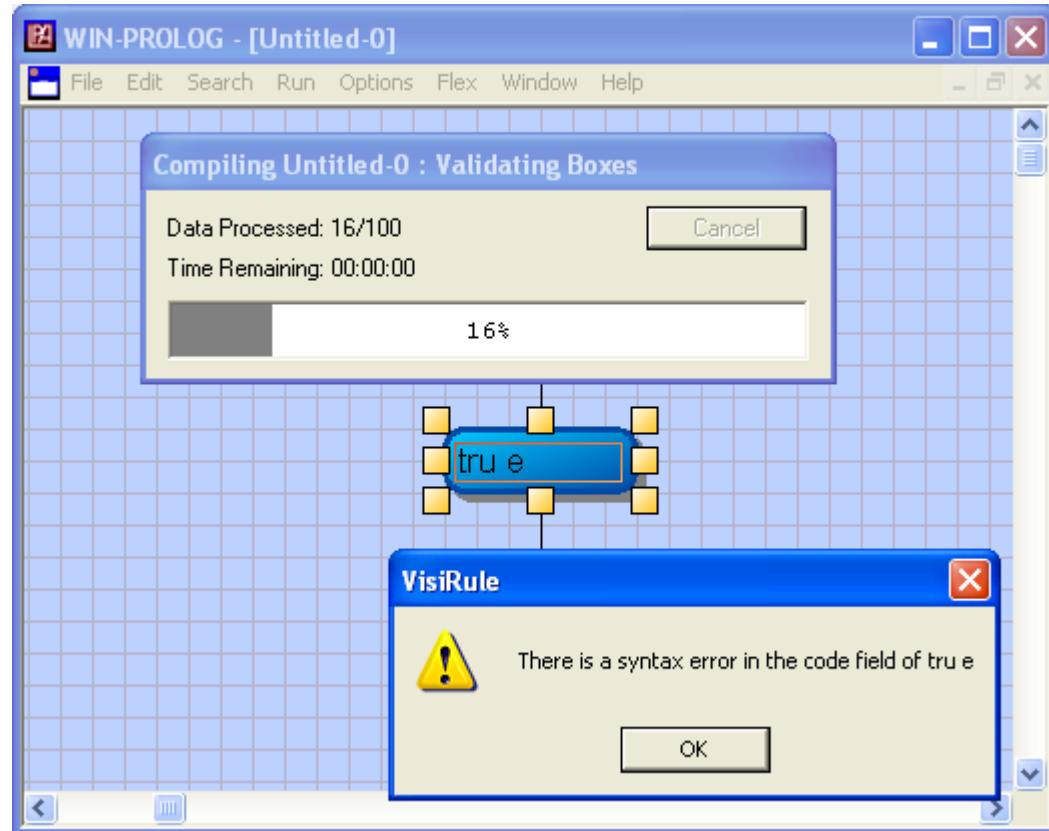


Figure 76 - Code box Syntax error

Syntax error in variable field of statement box

A dialog similar to the following may appear if a statement box is deemed to have its variables incorrectly specified

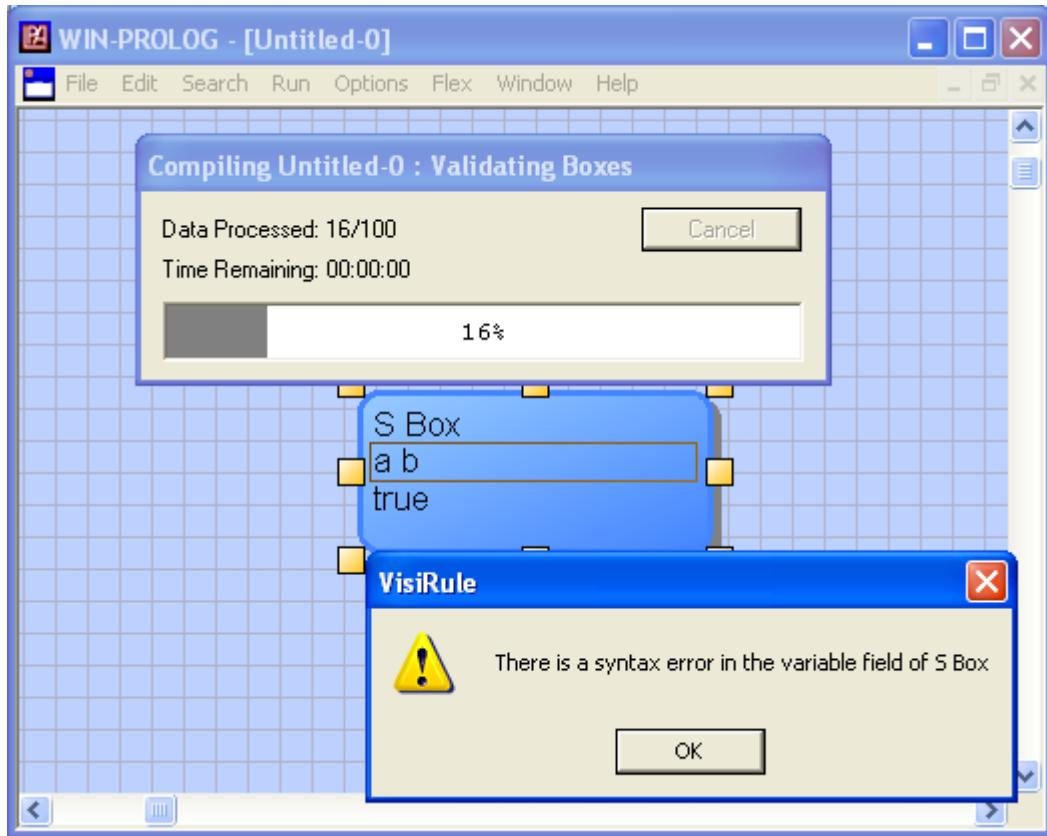


Figure 77 - Statement box Syntax error for variables

Syntax error in code field of statement box

A dialog similar to the following may appear if a statement box is deemed to have its code or Goal field incorrectly specified

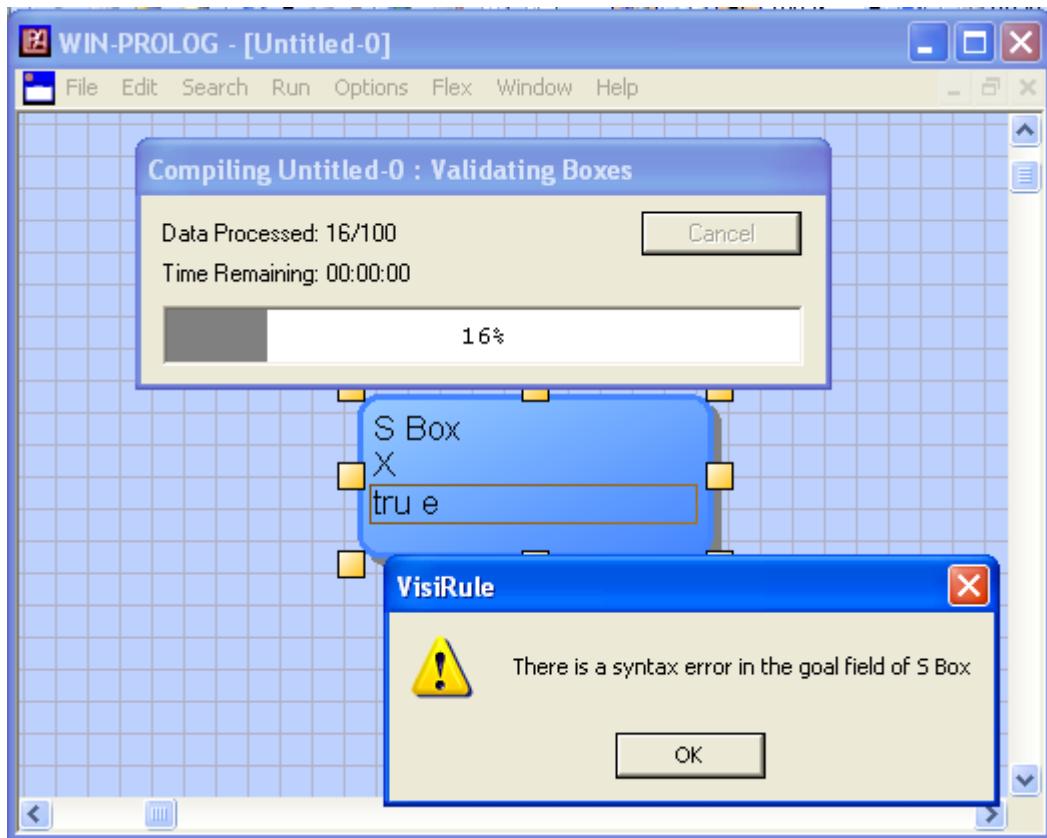


Figure 78 - Code box Syntax error in Goal or Code

Missing Reference

A dialog similar to the following may appear if you have a question which is not tested anywhere:

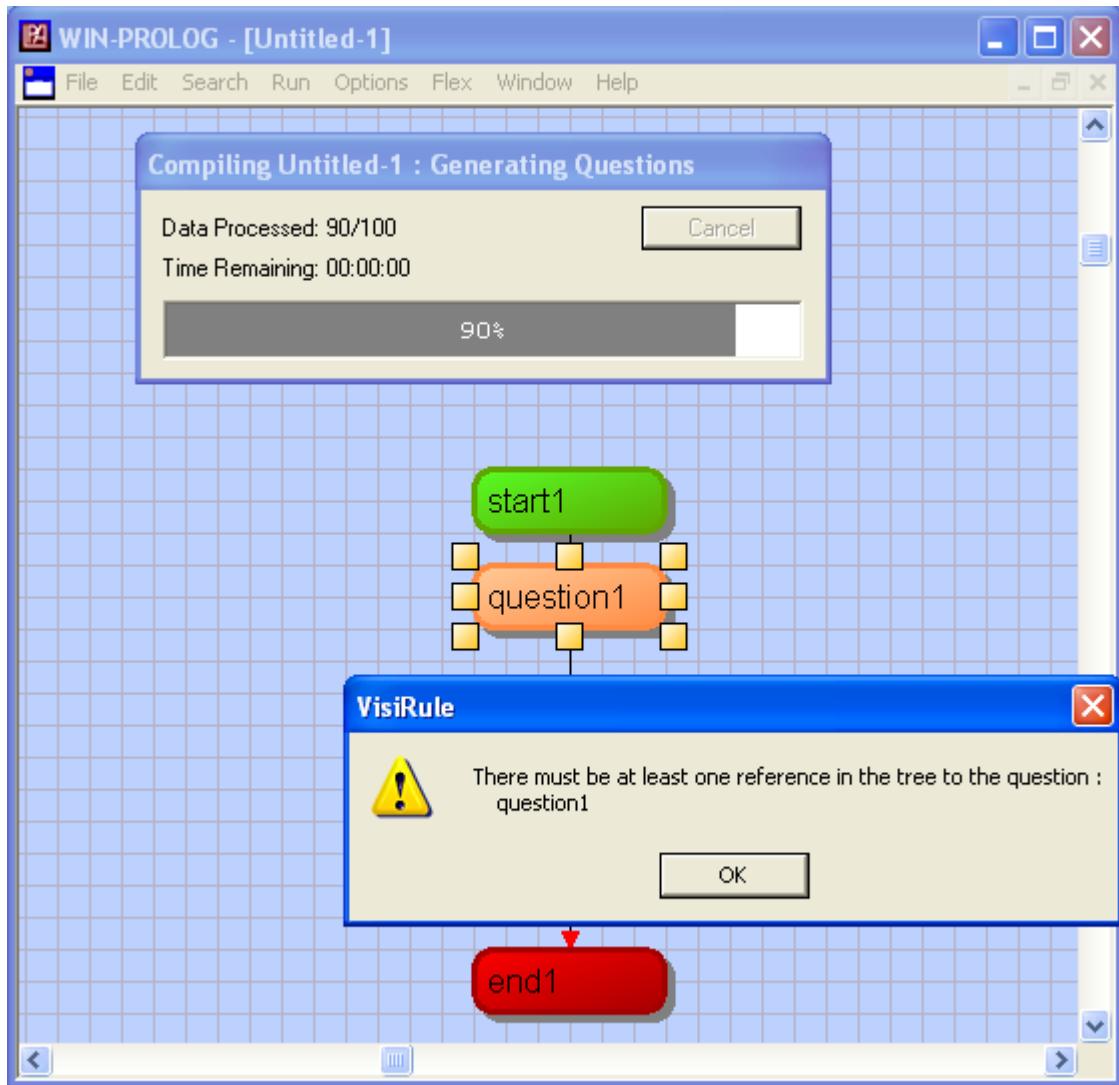


Figure 79 - Missing Reference error

'Analyse Why?' Flex Dialog

Sometimes, VisiRule may generate some Flex code, but the Flex compiler may have a problem compiling it. Remember, you can change the level of error reporting using the Prolog Flags dialog

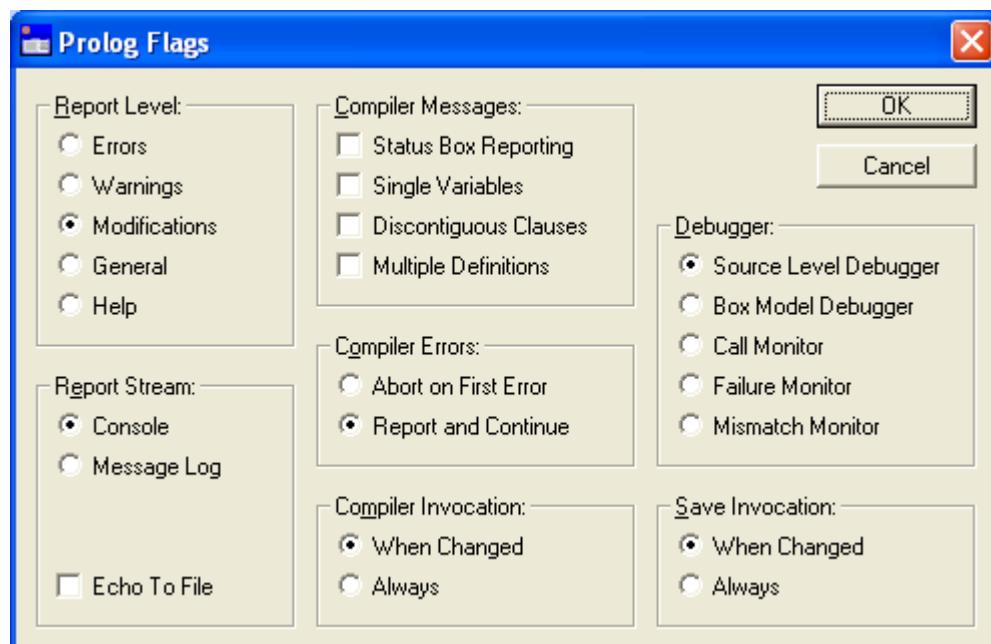


Figure 80 - Prolog Flags dialog

This may cause the 'Analyse Why?' flex dialog to appear. Click on the 'Yes' button to proceed with the analysis process.

The next dialog will indicate where the syntax error is to be found.

Syntax error in the region of... <text>

Continue analysis?

Click on the 'Yes' or 'No' button as required.

The syntax error itself is written out to the Console Window.

You can also copy the generated Flex code and paste it into a new Flex window and compile it using Run/Compile.

To return to your VisiRule worksheet window, click on the Window menu and select the appropriate entry.

Changing the Size of the Question Dialog

The size of any Question dialog can be changed via one of three settings in the [flex] section of the PRO386W.INI file:

```
[flex]
choose_one=200,100
```

The above will set all Single Choice Question dialogs to be 200 pixels wide by 100 pixels high. There are three settings - choose_one, choose_some and input:

INI File Settings	Question Dialogs Affected
choose_one	Single Choice
choose_some	Multiple Choice
input	Integer Input, Number Input, Set Input, Name Input

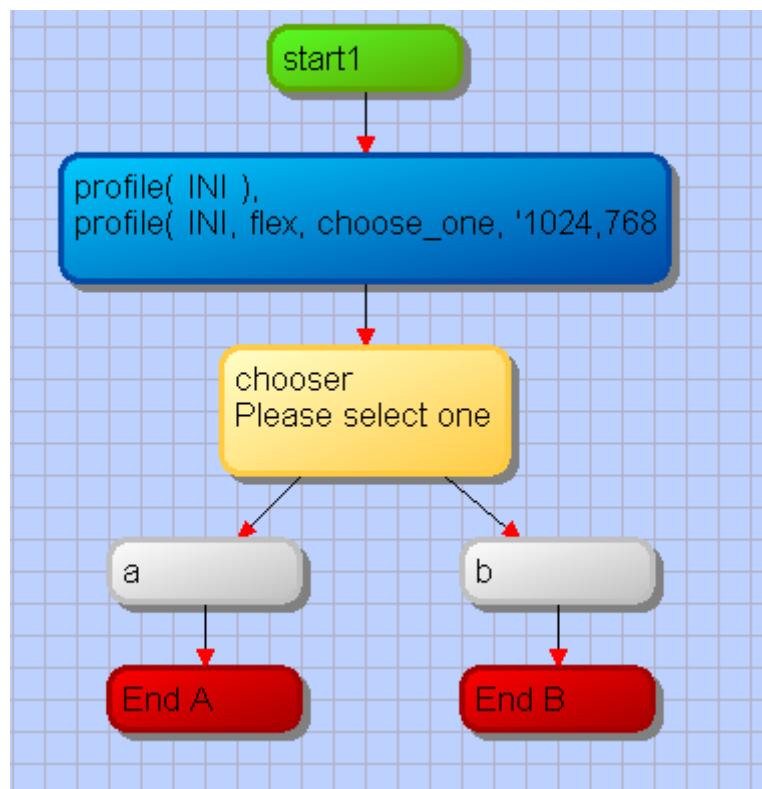


Figure 81 - Chart for resetting question sizes

You can change these settings programmatically using *profile/1* and *profile/4*:

The Debugging Process

Normal KSL Window

The generated code may be copied/pasted into a normal Flex KSL window. This means that it can then be compiled and debugged further using the standard LPA debugging tools and techniques that are provided for both Prolog and Flex code.

Displaying a Message or Outputting a Value

A flash command can be placed within a Statement or Code box to display a message or output a value whilst your chart is being executed. Please note that any message output using flash does not get stored.

The predicate, `write/1`, can be used as a means of logging values/messages to either the Console window or to a file.

Utilising Syntax Colouring

The code entered into a Statement or Code box is not syntax coloured. You could always write the code in a Prolog or flex edit window with on-the-fly syntax colouring, and then paste into the Statement or Code box when done.

APPENDIX A – ALLOWED LINKS

The table below shows which object types may be linked to which other object types:

APPENDIX B – INITIALISATION FILES

This appendix documents the settings within VISIRULE.CFG and PRO386W.INI relating to VisiRule.

VISIRULE.CFG

[system]

Variable	Value Format	Default Value	Comment
max_area		5,5	
grid	<float>,<float>	0.125,0.125	Determines the distance between the grid lines.
link_style		direct	
stack_size		50	
margin		0.5	
footer		0.4	
solid_markarea	<integer> in the domain {0,1}	0	Sets whether the marqui is solid (1) or not (0).
enclose_markarea	{0,1}	1	Sets whether an object is selected (0) or not (1) when half within a marqui.
show_grid	{0,1}	1	Sets whether the grid is shown (1) or not (0).
snap_to_grid	{0,1}	1	Sets whether objects are snapped to the grid (1) or not (0).
show_toolbar	{0,1}	1	Sets whether the toolbar is initially displayed (1) or not (0).
shadow	{0,1}	1	Show/don't show shadows on boxes
shadow_offset		3,3	
centerarrow	{0,1}	0	
arrowheads	{0,1}	1	Show/don't show arrowheads

boxpen		1	
boxfill		1	
nudgesize		1	
pixperinch		96,96	
roundboxcorner		15,15	
text_offset		5,5	
toolbar_metrics		4,4,24,24,3,5	
toolbar_position		16,90	
gradient		1	
arrowshape		$[(0,0), (3,-7), (-3,-7)]$	
epsilon		5	

[brush]

Variable	Default Value	Comment
vrl_statement	(153,204,255),(64,132,255),1,gradient	
vrl_single_choice	(255,255,204),(255,204,64),1,gradient	
vrl_multiple_choice	(255,204,153),(255,139,64),1,gradient	
vrl_number_input	(204,255,153),(131,201,0),1,gradient	
vrl_integer_input	(153,204,153),(0,150,32),1,gradient	
vrl_set_input	(255,204,204),(166,102,102),1,gradient	
vrl_name_input	(255,153,204),(166,0,97),1,gradient	
vrl_continue	(255,204,0),(166,86,0),1,gradient	
vrl_code	(0,204,255),(0,70,166),1,gradient	
vrl_start	(85,255,28),(94,166,0),1,gradient	
vrl_end	(255,0,0),(166,0,0),1,gradient	
vrl_expression	(255,255,255),(192,192,192),1,gradient	
vrl_arrow	255,0,0,solid	
vrl_selection	(255,255,255),(255,192,0),1,gradient	
vrl_elbow	255,192,0,solid	
vrl_shadow	128,128,128,solid	
vrl_background	190,210,255,solid	
vrl_toolbar	(255,255,255),(170,170,170),1,gradient	
vrl_arrow	255,0,0,solid	

[pen]

vrl_statement	64	132		2 .
vrl_single_choice	255	204	64	2 .
vrl_multiple_choice	255	139	64	2 .
vrl_number_input	131	201	0	2 .
vrl_integer_input	0	150	32	2 .
vrl_set_input	166	102	102	2 .
vrl_name_input	166	0	97	2 .
vrl_continue	166	86	0	2 .
vrl_code	0	70	166	2 .
vrl_start	94	166	0	2 .
vrl_end	166	0	0	2 .
vrl_expression	192	192	192	2 .
vrl_box	64	64	64	2 .
vrl_drag_object	0	0	0	dot .
vrl_shadow	128	128	128	2 .
vrl_arrow	255	0	0	solid .
vrl_grid	180	180	200	solid .
vrl_hilite	255	255	255	solid .
vrl_lolite	128	128	128	solid .
vrl_selection	0	0	0	solid .

[fore]

Variable	Default Value	Comment
vrl_font	'Arial',14,0	
vrl_error	255,0,0	

PRO386W.INI

[flex]

Variable	Sample Value	Comment
choose_one	1000,100	Alters the size of a 'choose_one' (i.e. Single Choice) question dialog.
choose_some	100,1000	Alters the size of a 'choose_some' (i.e. Multiple Choice) question dialog.
input	100,100	Alters the size of an 'input' (i.e. Integer Input, Number Input, Set Input, Name Input) question dialog.

Index

.	
.KSL	62
.vsr	24
.WMF	62
<	
< 78, 79	
=	
= 78, 79	
=<	78, 79
>	
> 78, 79	
>=	78, 79
A	
Align Submenu	55
Analyse Why?	103
and.....	82
B	
Box Type Submenu	48, 52
Box Type tab	57
Box Types	68
Brackets	83
C	
Clear Option	53
Code Box	86
Code Generation Errors	97
Combination Operators	82
Comment Box	88
comparison operators	81
Comparison Operators	78
Compiling Your Chart	93
Complex Expressions	83
Console window	86
Continue Box	89
Copy Option	53
Creating an Object	29
Ctrl + A	66
Ctrl + Alt + Spacebar	66
Ctrl + C	66
Ctrl + Spacebar	66
Ctrl + V	66
Ctrl + X	66
Ctrl + Y	66
Ctrl + Z	66
Cut Option	52
D	
Delete.....	66
E	
Delete Option	57
Deleting Objects	36
Deselecting Objects	47
F	
Edit Time Errors	96
Editing Text	33
End Box	90
Enhanced Windows Metafile	62
equal to	78
Errors	96
Errors and Debugging	96
Escape	67
Executing Your Chart	93
Export... Option	62
Expression Box	76
G	
Field Visibility tab	57
File Menu	59
flash Command	86
flash/1	86
flex.....	62, 85, 86
flex code	93
H	
Generated Code Dialog	93
greater than	78
greater than or equal to	78
I	
Incomplete Chart	98
I nteger Input	73
Integer Input Question Box	75
J	
Java	94
K	
Keyboard Shortcuts	66
L	
less than	78
less than or equal to	78
Linking Objects	36
Links	106
M	
Metafile	62
Missing Reference	102
More Smart Linking	45
Moving Objects	35
M ultiple Choice	73
Multiple Choice Question Box	75

N	
Name Input	73
Name Input Question Box	75
Naming a Box	68
Navigation	47
New... Option	59
not equal to	78
Number Input	73
Number Input Question Box	75
O	
Object Menu	51
Open... Option	59
Options Menu	65 or 82
P	
Paste Option	53
precedence	83
PRO386W.INI	112
Prolog code	85, 86
Properties dialog	57
Properties Option	57
Q	
Question Boxes	70
Question name	87
R	
Redo Option	50, 52
resize question dialog	112
S	
Save Option	60
Set Input	73
Set Input Question Box	75
set notation	81
Shift + Enter	67
Shift + Tab	67
Show All Code... Option	66
Show Code... Option	50, 66
Show Grid Option	49, 65
Show Toolbar Option	65
Single Choice	73
Single Choice Question Box	75
T	
Tab	67
toolbar	25
U	
Undo Option	49, 52
Unlinking Objects	38
V	
VB94	
VISIRULE.CFG	107
W	
WebFlex	94
When to Ask Questions	91
Windows Metafile	62
WIN-PROLOG development environment	23
Working with Boxes	30
worksheet	24
Worksheet Menu	48
X	
XML	62
Z	
Zoom Submenu	49
Zooming In	47
Zooming Out	47