Quantum Computing with VBA

By Tamás Bunth
Software Engineer at Collabora Productivity
VBA and ...

Teaching people radical new ways to model problems

- Great simulation tools for training
- Fronted by an easy-to-use Quantum Apprentice.xlsm
  - VBA / Macro-enabled Excel spreadsheet document
  - Visualization and interface to backend D-Wave simulator for training.
  - Dynamically generated elements in the document.

Problem:

- Difficult to deploy & publish widely
- Depends on Microsoft Office, installation & setup is tricky
- VMs a distribution challenge
Quantum Computing

- World’s first quantum computing company
- Quantum computing systems and software
  - Built around “qubits” rather than “bits”
  - Revolutionary technology leaders creating a new industry.

“Collabora understood our problem immediately, and implemented a great, Open Source solution for us to improve customer training” - Denny Dahl, Principle Research Scientist at D-Wave Systems Inc.
Tabs

- Two Qubits, Three Qubits, Four Qubits: static content
- Chimera tab
- QMI, Solution
- Problem History tab
<table>
<thead>
<tr>
<th>Q0071</th>
<th>Q0072</th>
<th>Q0073</th>
<th>Q0074</th>
<th>Q0075</th>
<th>Q0076</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q0087</td>
<td>Q0088</td>
<td>Q0089</td>
<td>Q0090</td>
<td>Q0091</td>
<td>Q0092</td>
</tr>
<tr>
<td>7.313432859</td>
<td>3.805970013</td>
<td>9.179104149</td>
<td>7.313432859</td>
<td>3.507462632</td>
<td></td>
</tr>
<tr>
<td>Q0103</td>
<td>Q0104</td>
<td>Q0105</td>
<td>Q0106</td>
<td>Q0107</td>
<td>Q0108</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q0119</td>
<td>Q0120</td>
<td>Q0121</td>
<td>Q0122</td>
<td>Q0123</td>
<td>Q0124</td>
</tr>
</tbody>
</table>
Chimera Tab
<table>
<thead>
<tr>
<th>QMI name</th>
<th>Solution name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-qubit-chain</td>
<td>chain</td>
</tr>
<tr>
<td>bar</td>
<td>facet</td>
</tr>
<tr>
<td>default</td>
<td>result</td>
</tr>
<tr>
<td>test1</td>
<td>ring</td>
</tr>
</tbody>
</table>

**Problem History Tab**

- Several buttons
- List of QMI files, Solution files
- Select workspace, refresh, clear, etc.
Demo
Errors with LibreOffice?

- Nothing happened first
- Only static elements displayed
  - Workbook_Open()?
- Let’s see something interactive
  - Refresh History button → Exception: property not found.
Workbook_Open()

- Workbook.Open event
- Set default workspace
- Read geometry
- Draw Chimera, draw QMI, draw Solution
- Refresh history (qmi files, solution files)
Private Sub Workbook_Open()
    ' 20 lines: ' Initialize the variable that tells
    Call ReadGeometry(initial_geometry)
current_configuration = 1
    Call Update_DW_INT
    Call Write_dwsync
    Call Init

    Application.ScreenUpdating = False
    Call DrawChimera
    Call DrawQMI
    Call DrawSolution
    Call GetHistory
    Worksheets("Two Qubits").Activate
    Application.ScreenUpdating = True
End Sub
Workbook.Open event

Register / trigger

- Register → Ok
- Trigger → Exception thrown

EnsureVBA.Library

- Only with .xlsm extension
- “BasicLibraries” not created
Back to Refresh History

Missing properties

- Interior.TintAndShade
- Interior.PatternTintAndShade

ScVbaInterior::GetPatternColorIndex()

- Used in the opposite way
- Probably never ever used before
Sub GetHistory()
    ' This macro is invoked by pressing the "Refresh History" button on the
    ' Problem History worksheet
    Dim varDirectory As Variant
    Range(Cells(2, 1), Cells(iqmi, 1)).Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = -0.15
        .PatternTintAndShade = 0
    End With

    Range(Cells(2, 3), Cells(isol, 3)).Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = -0.15
        .PatternTintAndShade = 0
    End With

    Worksheets("Problem History").Range("I19").Select
End Sub
<table>
<thead>
<tr>
<th>QMI name</th>
<th>Solution name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-qubit-chain</td>
<td>chain</td>
</tr>
<tr>
<td>bar</td>
<td>facet</td>
</tr>
<tr>
<td>default</td>
<td>result</td>
</tr>
<tr>
<td>test1</td>
<td>ring</td>
</tr>
</tbody>
</table>
Determine OS

- Used in the macro for determining path type.
- Application.OperatingSystem (read-only) property
- Implement using preprocessor directives.

\[
\begin{align*}
\text{TheOS} &= \text{Application.OperatingSystem} \\
\text{DWAVE\_HOME} &= \text{Replace(Environ(\text{"DWAVE\_HOME"}), \\
&\quad \text{"/"}, \\
&\quad \text{Application.PathSeparator})} \\
\text{If left(TheOS, 3)} &= \text{"Mac"} \text{ Then} \\
&\quad \text{DWAVE\_HOME} = \text{"Macintosh HD"} + \\
&\quad \text{DWAVE\_HOME} \\
\text{End If}
\end{align*}
\]
Drawing the ellipses

Finally drawing the Chimera tab

- To visualize qubits.
- GetType() does not know Oval (Ellipse)

```
Worksheets("Chimera").Shapes.
AddShape(msoShapeOval, ... )
```
Chimera buttons

- View QMI, Random, Reset
- Characters.Font.Superscript / Subscript properties
  - RuntimeException
- Attributes in form controls
  (WrappedTargetRuntimeException)

Reading order

XlContext

- Depends on first character
- Left-to-right or right-to-left
- Default with Excel

Not implemented

- NotImplementededException vs. left-to-right
Choose workspace

Button in Problem History tab

- Application.FileDialog ↔ XFilePicker, XFolderPicker
- ScVbaFileDialog, ScVbaFileDialogItems
- Show(), InitialFileName
- System path vs. URL → osl::FileBase
Select Workspace
Function Select_Workspace_Win()

    With Application.FileDialog(msoFileDialogFolderPicker)
        .InitialFileName = DW_SIM_DIR & "\\"
        .Title = "Select a workspace directory"
        .Show
    If .SelectedItems.Count = 0 Then
        Select_Workspace_Win = ""
    Else
        Select_Workspace_Win = .SelectedItems(1)
    End If
End With
End Function
Slow...

- Valgrind (kcachegrind)
- Avoid AdjustRowHeight() calls
  - When applying attributes
  - When screenUpdating is false
TypeOf myShape Is Oval

- TypeOf ... Is ... operator
  - In ‘basic/...’, common basic operator
- Actual result → “shape”
  - XtypeProvider::getTypes()
TypeOf

Solution

- ImplInheritanceHelper
- XOval, Xline ← oovbaapi
- ScVbaOvalShape, ScVbaLineShape ← sc/source/...
Thank you!

By Tamás Bunth
IRC: wastack
Email: tamas.bunth@collabora.co.uk