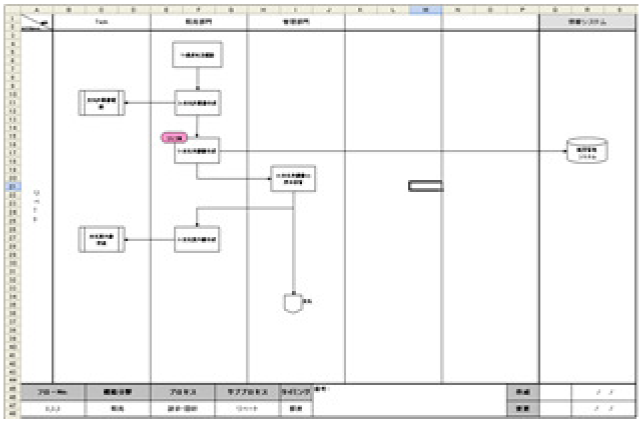
Excel AutoShapes Plugin

*The **KT.AM.Import.ExcelAutoShapes** plugin allows you to import legacy Microsoft Office AutoShape based flowcharts as fully compliant BPMN process diagrams.*

About the AutoShapes Plugin

This plugin will convert Excel worksheets containing Microsoft Office AutoShape flow diagrams into standard BPMN.

A typical Excel flow diagram

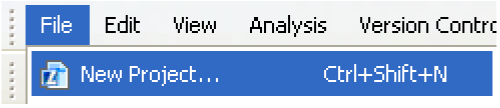


Excel flowcharts commonly use vertical lines to specify divisions of domain. The converter will create swimlanes for these divisions

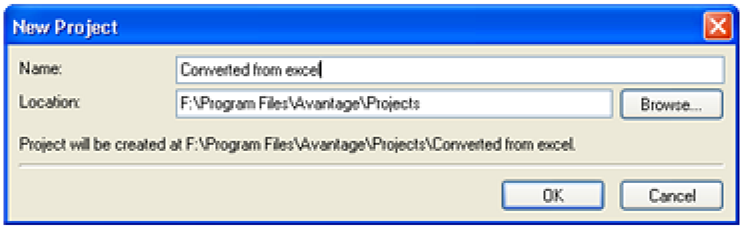
How to import

Create a project

Avantage organises your work into projects. Projects contain related files (like process models, categories, organisations, etc). To create a new project choose the *New project..* menu from the *File* menu.



Enter the name of your new project into the new project dialog. Don't change the location - just provide a name.



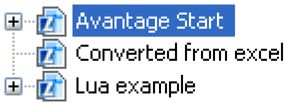
New Project

Name: Converted from excel

Location: F:\Program Files\Avantage\Projects

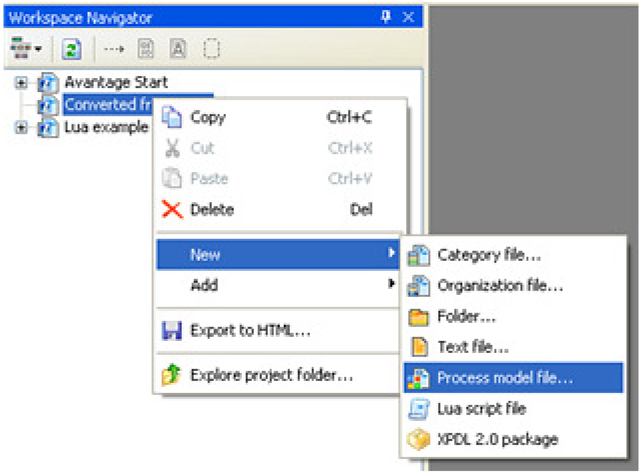
Project will be created at F:\Program Files\Avantage\Projects\Converted from excel.

After you have done this your workspace navigator should look like this:

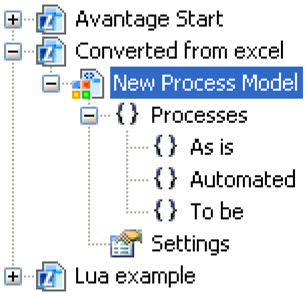


Create a process model to contain the diagrams

Right-click on your freshly minted Excel project and choose the *New:Process model file...* action.



Expand the new process model, it will look like this:

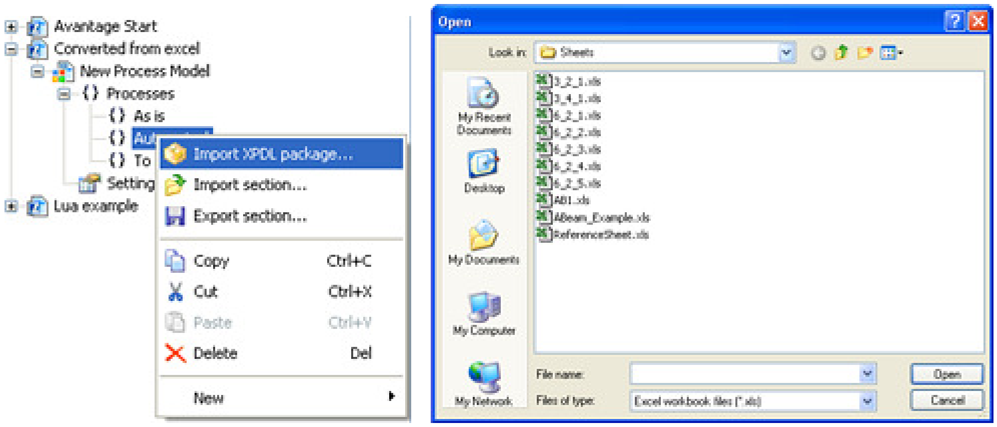


The nodes in the tree named *Processes, As is, Automated* and *To be* are what we call MeasurePoints.

Run the convertor

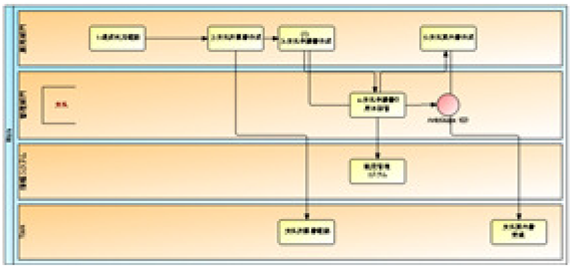
Right-click on a *measurepoint* that is to become the parent of the imported diagram and choose the *Import diagram from Excel worksheet...* action. You are prompted to choose a file or selection of files to convert.

The converted diagram is added at the measurepoint you selected. You can explore its structure by clicking on the diagram's + sign or you can open it in the diagram editor by double-clicking it.



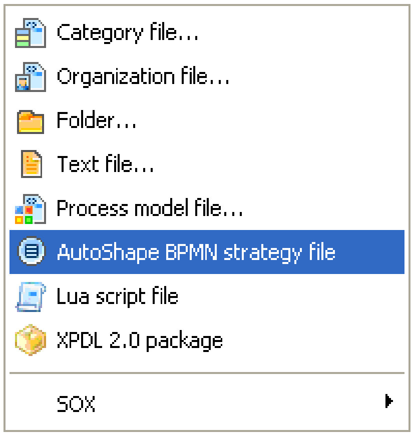
Geometric layout of converted BPMN diagram

X and Y coordinate information from the Excel flowchart is discarded by the convertor and instead it relies on the Diagram Editor to perform a once only automatic layout the first time the diagram is opened. You can then edit the diagram manually and change the layout/routing of connections as necessary.

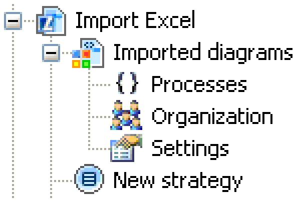


How to customize the shape conversion

Create a new strategy file by right-clicking on your project's icon and choosing the **New:Autoshape BPMN strategy file** action.



This will create an entry in project like this:



The new file will inherit the settings of the current strategy defined in the Preferences page.

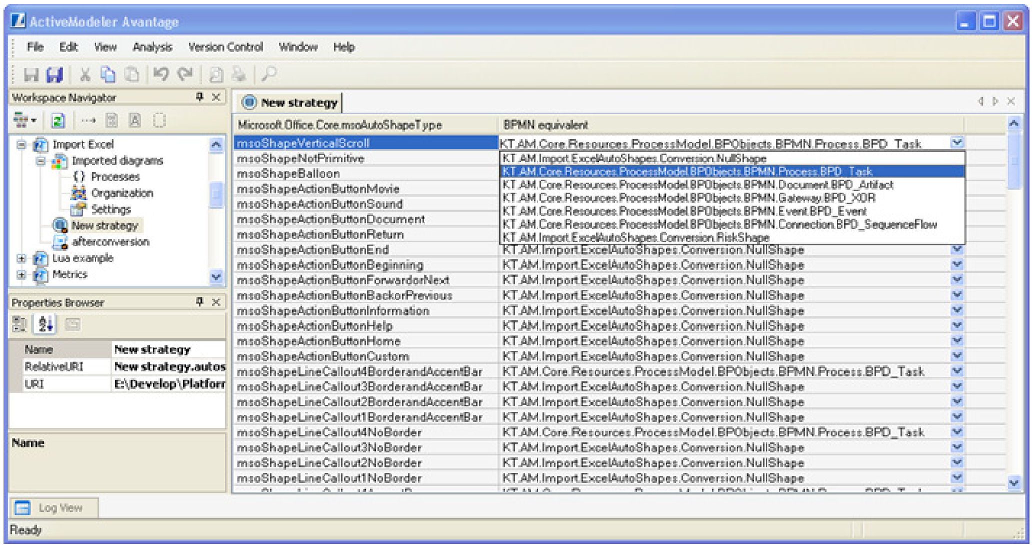
How to set a default strategy file

You can set the default strategy file (which will over-ride the preferences page) by right-clicking on a strategy file and choosing the “Set as default autoshape strategy” action.

The icon will be displayed with a tick mark to indicate its status as the default strategy. You can turn it off by right-clicking and choosing the “Remove as default autoshape strategy” action.

Editing a strategy file.

Double-click on thestrategy icon in the navigator. This will display the strategy editor window.



Name	New strategy
RelativeURI	New strategy.autos
URI	E:\Develop\Platforr

Name

Microsoft.Office.Core.msoAutoShapeType	BPMN equivalent
msoShapeVerticalScroll	KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Process.BPD_Task
msoShapeNotPrimitive	
msoShapeBalloon	
msoShapeActionButtonMovie	
msoShapeActionButtonSound	
msoShapeActionButtonDocument	
msoShapeActionButtonReturn	
msoShapeActionButtonEnd	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonBeginning	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonForwardorNext	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonBackorPrevious	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonInformation	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonHelp	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonHome	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeActionButtonCustom	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout4BorderandAccentBar	KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Process.BPD_Task
msoShapeLineCallout3BorderandAccentBar	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout2BorderandAccentBar	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout1BorderandAccentBar	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout4NoBorder	KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Process.BPD_Task
msoShapeLineCallout3NoBorder	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout2NoBorder	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
msoShapeLineCallout1NoBorder	KT.AM.Import.ExcelAutoShapes.Conversion.NullShape

KT.AM.Import.ExcelAutoShapes.Conversion.NullShape
KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Process.BPD_Task
KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Document.BPD_Artifact
KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Gateway.BPD_XOR
KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Event.BPD_Event
KT.AM.Core.Resources.ProcessModel.BPObjects.BPMN.Connection.BPD_SequenceFlow
KT.AM.Import.ExcelAutoShapes.Conversion.RiskShape

The left column contains a list of all of the AutoShape types the plugin can recognize.

The right column indicates the BPMN object that it will be converted to which you can change by operating the drop-down menu of BPMN object types.

Custom post-conversion processing

You can tell the reader plugin to execute a Lua script at the completion of the conversion. This makes it possible to scrape other information from the worksheet that may be relevant to the process model you are creating. For example, the worksheet might contain metrics (volumes, costs etc) and a post-conversion script could read this information and write it into the process model.

How to specify a default post-conversion script

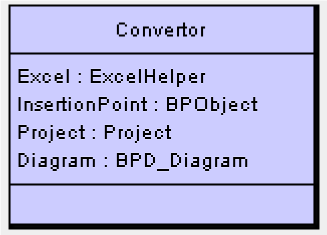
A prototype (template) script is provided in the `Import.Excel.AutoShapes` plugin folder. It is named `afterconversion.lua`. You can add this file to your working project with the *Add existing Lua file...* context menu (a copy of this file will be added into your project). The file should now be displayed in the project's sub-tree. Right-click on it and choose the *"Set as default post conversion script"* menu action. This will make this file the new default .lua file which runs post-conversion. To turn the default off, right-click on the file again and choose the *"Remove as default post conversion script"* menu action.
You can easily see which script is the default post conversion for a project because its icon is overlayed with a tick mark like the image below.

Passed in parameters

The script receives a Convertor object and a Progressmonitor object.

The Convertor object

The Convertor object contains a reference to the conversion context.



Excel

This is a reference to a simplified .NET Excel wrapper.

InsertionPoint

This is the BPObject which represents the point in the BPMNTree where the Diagram is to be inserted. It is usually a BPD_MeasurePoint type object.

Project

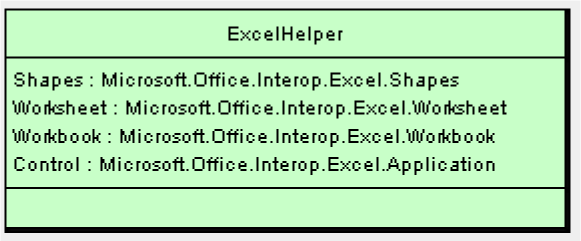
This is the Avantage project in which the action is taking place.

Diagram

This is the BPD_Diagram object that was created by the convertor.

The ExcelHelper object

The ExcelHelper holds references to the .NET interop that is managing the Excel COM instance.



```
ExcelHelper
Shapes : Microsoft.Office.Interop.Excel.Shapes
Worksheet : Microsoft.Office.Interop.Excel.Worksheet
Workbook : Microsoft.Office.Interop.Excel.Workbook
Control : Microsoft.Office.Interop.Excel.Application
```

Changing autoshape BPMN conversion rules

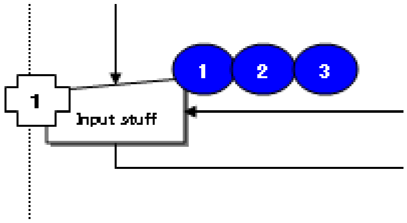
You can change BPMN shape lookup tables using a table editor. Operate the drop-down menu in the BPMN column to select the BPMN object to be created. *NullShape* is a special do nothing shape.

Importing metrics and documentation references

A spreadheet is likely to contain numerical data which is sometimes associated with flow objects, for example costs may be assigned to various activities. To do this you will need to purchase the *Avantage Analytics Plugin pack*. (See Analytics Pack for more details). The Analytics Pack provides a business metrics DFD mixin for BPMN diagrams.

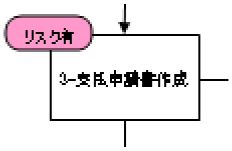
SOX metadata in the Excel flow chart.

Figure 1. SOX indicators used in the msoShapeFlowChart template



The blue shapes represent **risk points**. The white cross shape represents control information about the activity.

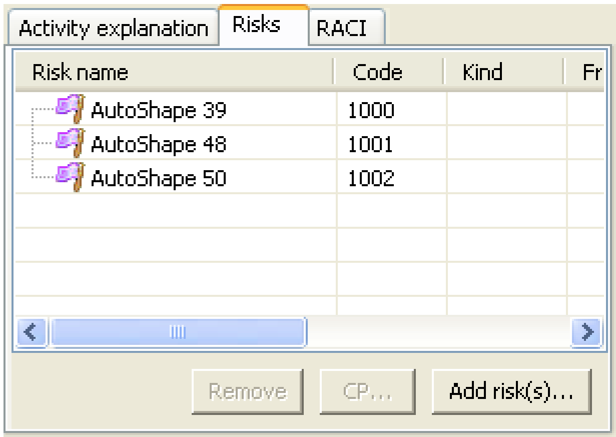
Figure 2. Sox indicators used in the ABeam template



The plugin will ignore risk points which do not overlap. One of them must overlap the 'parent' task shape as well.

In the figure above, all of the risk shapes and the white cross shape overlap which will enable SOX information to be recorded for the BPD_Task object which has been created for “Input stuff”.

Figure 3. SOX inspector showing the blue oval shapes converted to risk points

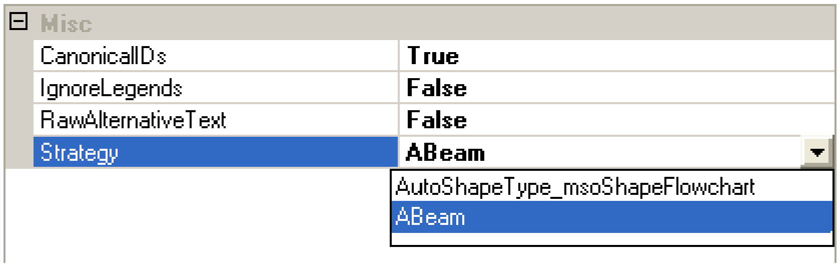


Risk name	Code	Kind	Fr
AutoShape 39	1000		
AutoShape 48	1001		
AutoShape 50	1002		

Preferences

The Preferences page lets you choose between templates strategies and some other settings.

Figure 4. The plugin preferences



Misc

CanonicalIDs	True
IgnoreLegends	False
RawAlternativeText	False
Strategy	ABeam

Canonical Ids

This option is currently not used.

Ignore Legends

Some templates have a “Legend” lane which contains a key of the shapes used in the flow chart.   This information is probably not worth keeping, since BPMN defines the notation. By default the plugin keeps this information.

Raw alternative text

When true, the contents of an autoshape's Alternative text property is used to set the BPMN object's name property. When set to false, just the user entered portion of the text is set as the BPMN object's name.

Strategy

This setting chooses the template. It is called a strategy because a template could be interpreted differently depending upon the intent of the conversion.There are currently two template strategies defined based on what is currently being used in Japan.

msoShapeFlowChart:

This template standard uses the traditional flowchart symbols to represent a process. Swimlanes are depicted by Excel border columns.

The first column of an msoShapeFlowChart flow is a legend column containing the symbols used in the flow chart. The conversion process puts these into a Swimlane called [1].

This template is used to describe SOX risks. The conversion process maps them into SOXCOSO RiskPoints

ABeam

This template standard uses a slightly different representational format. It is very similar to the msoShapeFlowChart style except that a different set of symbols used used to represent flow objects.

Appendix A: Conversion of name properties

When composing a BPMN object name the plugin will give first preference to the user entered portion of the AutoShape.AlternateText property (if it is present), otherwise the Autoshape.Name property is used.

The AlternateText property of a typical autoshape object will contain a string in the form of “Flowchart: Manual Input: The following text entered by the user”.

The plugin extracts just the user entered information to the right of the last “:” character in the AlternateText property.

In case the full alternate text is important, the plugin writes it to the BPMN .Documentation property of the created BP object.

Appendix B: Post-conversion stub script

```
---
--- This script is called after the Excel Autoshape conversion
has converted the Excel shapes collection
--- into BPMN.   It is an afterburner type script template
which can be used to perform custom
--- post processing of the Excel workbook after the BPMN model
has been created.
---

luanet.load_assembly("Microsoft.Office.Interop.Excel")
Range =
luanet.import_type("Microsoft.Office.Interop.Excel.Range")
MessageBox = luanet.System.Windows.Forms.MessageBox

--- There are two variables defined by the plugin just before
it calls this script.
---     1. Monitor which is an IProgressMonitor object
---     2. Convertor which is a
KT.AM.Import.Excel.AutoShapes.ToBPMN convertor object. This
contains a reference
---     to the Excel instance being worked with, it also
provides the BPMNTree output context for the conversion.
```

```
--- This is how to update the progress display during a
repetitive operation
for i = 0, 3000, 1 do
        Monitor.SubTaskName = "Script loop " .. i
        --- Do something
end
return 0
```

Conversion Issues

Excel document validity

An Excel flowchart may look OK without it necessarily being correct.

Sometimes the directed graph in the Excel flowchart is broken and it must be fixed before you can sucessfully import the data.

Connecting lines must be properly connected at both ends between shapes in the source Excel sheet. In the development of this plugin, it was discovered that what visually appeared to be correct was in fact a broken link in the graph which prevented the plugin from constructing a BPMN sequence flow.

The problem is a connector line which is not docked to a port on another shape at both ends. You can easily diagnose this problem by dragging the shape in Excel to see if all of its connectors redraw themselves at the new location. If they don't relocate themselves then you will have to re-dock the offending line(s) to shape(s) before proceeding with the conversion.