



# ActiveModeler Avantage



# Foundation Users Guide

© 2006, 2007, 2008 KAISHA-Tec Co. Ltd. Japan

Sangyo Plaza Annex 3-32-3 Shimo Renjaku,  
Mitaka-shi Tokyo, 181-0013 JAPAN

Document Revision:	1.60
Date:	February 2, 2008
Printed	11:30:44 AM

ActiveModeler™ and ActiveFlow™ and ActiveModeler Avantage™ are registered trade marks of KAISHA-Tec Co. Japan

# Acknowledgements

KAISHA-Tec would like to thank Steve White of IBM for permission to include material from his introductory BPMN material in this guide.

# Copyright Notice

© 2006, 2007, 2008 KAISHA-Tec Co. Ltd Japan. All rights reserved.

ActiveModeler and ActiveFlow are registered Trade Marks of KAISHA-Tec Co. Ltd Japan.

# Legal Notice

BPML.org require that this notice be contained in documents that refer to and or contains material derived from the Business Process Modeling Notation (BPMN) Specification.

## *Notice of BPML.org Policies on Intellectual Property Rights & Copyright*

*BPML.org takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on BPML.org's procedures with respect to rights in BPML.org specifications can be found at the BPML.org website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the BPML.org Chairman. BPML.org invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights, which may cover technology that may be required to implement this specification. Please address the information to the BPML.org Chairman.*

*This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to BPML.org, except as needed for the purpose of developing BPML.org specifications, in which case the procedures for copyrights defined in the BPML.org Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.*

*The limited permissions granted above are perpetual and will not be revoked by BPML.org or its successors or assigns.*

*This document and the information contained herein is provided on an "AS IS" basis and*

***BPML.org DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.***

*Copyright © The Business Process Management Initiative [BPML.org], May 3, 2004. All Rights Reserved.*

## Table of Contents

Introducing the ActiveModeler Avantage BPM Platform.....	19
About ActiveModeler Avantage.....	19
Features and benefits.....	20
Quick Introduction to the architecture of ActiveModeler Avantage.....	21
Installation.....	22
Prerequisites.....	22
Installing the .NET Framework 3.0.....	22
How to determine which version of the .NET Framework is installed on your computer.....	22
Running Setup.....	23
Multi lingual deployment.....	24
Adding plugin licenses.....	25
An introduction to BPMN.....	27
Why BPMN?.....	27
What Is BPMN?.....	28
BPMN Basics.....	29
Flow Objects.....	29
(i) Event.....	29
(ii) Activity.....	30
(iii) Gateway.....	30
Connecting Objects.....	31
(i) Sequence Flow .....	31
(ii) Message Flow.....	32

(iii) Association.....	32
Allowable Sequence flow connections.....	33
Allowable Message flow connections.....	34
Swimlanes.....	36
Artifacts.....	38
General uses of BPMN.....	41
Internal business processes.....	41
Collaborative B2B Processes.....	41
Different levels of precision.....	42
Mapping a BPMN diagram to Execution Languages.....	44
XPDL 2.0 Plugin.....	44
The Road ahead.....	45
Concepts and terminology.....	46
Workspaces.....	46
Projects.....	47
Files.....	47
Views.....	47
Editors.....	48
Actions (commands).....	48
Plugins.....	48
Simple configuration.....	49
QuickStart Guide.....	50
The first time you run Avantage .....	50
Working with docking windows.....	53
To move and re-dock a docked window.....	53

Floating views .....	57
Tabbed views .....	58
Pinned views (auto-hide).....	59
A suggested layout.....	61
Preferences.....	62
How to Create a Project .....	63
How to create a Process Model.....	67
Renaming a process model.....	67
Process model options.....	68
Navigator Toolbar Buttons.....	70
Object actions.....	71
Navigator Symbols.....	72
To create a process diagram.....	74
Placing flow objects in a diagram.....	77
Changing a Shape's properties.....	81
Saving and coordinating changes.....	83
Saving files at exit.....	84
The Save action.....	85
For Diagram windows.....	85
For the Workspace Navigator window.....	85
Forcibly saving a process model.....	85
The Save As action.....	85
For Diagram windows.....	85
For Workspace Navigator .....	86
Process model Revert.....	86

How to Revert to a previous generation.....	86
The different types of modifications/changes.....	88
Structural changes.....	88
Drawing canvas changes.....	88
Clipboard operations .....	89
Navigator (BPTree) clipboard operations.....	89
Paste and paste targets.....	90
Copying objects from one process model to another.....	91
Copying objects from one part of a process model to another.....	91
Process flow connecting objects.....	92
Sequence flows.....	92
Message flows.....	92
Associations.....	92
How to create connection objects.....	93
Creating connections with the BPMN toolbar.....	93
Creating connections using context menus.....	93
Nested Process Descriptions (Sub maps).....	94
Independent (external) sub processes.....	95
Embedded (internal) sub processes.....	96
The Diagram Editor.....	98
Diagram editor windows.....	98
The Canvas.....	99
To change the size of your canvas.....	99
The Diagram Menu.....	100
Rulers.....	100



Selections.....	100
Selecting Pools, Lanes and expanded SubProcess shapes.....	101
Moving things around.....	102
Changing the size of shapes.....	102
Changing label text.....	102
Editing sequence flow visual route.....	103
Changing the vertical or horizontal position of a line segment.....	104
Adding a turn (vertex) to a connection.....	104
Re-linking connections.....	104
Shape actions.....	106
Show labels.....	106
Shape properties.....	106
.....	107
Auto layout.....	108
Enabling Auto layout.....	108
Forcing re-layout.....	108
Moving pools and lanes.....	109
Layout Preferences.....	111
Auto layout triggers.....	111
After deletion.....	111
After inserting a shape.....	111
After linking.....	111
Lane ordering.....	111
Sort lane order.....	112
Padding.....	112

Pool left padding.....	112
Pool padding.....	112
SubProcess padding.....	112
Swimlane padding.....	112
Swimlane left padding.....	112
Sizes.....	112
Activity horizontal space.....	112
Activity vertical space.....	112
Minimum expanded SubProcess size.....	112
Minimum pool size.....	112
Navigating within a large process diagram.....	113
Using the Overview window.....	113
The Navigate actions.....	114
Left edge.....	114
Right edge.....	114
Page left.....	114
Page right.....	114
The BPMN toolbar.....	149
How to use the BPMN toolbar.....	149
Screen snapshot tool.....	149
Activity tools.....	149
Add Task.....	149
Add SubProcess.....	150
Gateway tools.....	150
Add XOR gateway.....	150

Add OR gateway.....	150
Add AND gateway.....	151
Add Complex gateway.....	151
Event tools.....	151
Add Start event.....	151
Add Intermediate event.....	152
Add End event.....	152
Adding an event trigger to an event.....	153
Swimlane tools.....	154
Add Pool.....	154
Add Lane.....	154
Artifact tools.....	155
Data object artifact.....	155
Text annotation artifact.....	155
Displaying document information.....	155
Grouping artifact.....	159
.....	159
Linking tools.....	160
Add Message flow.....	161
Add Association.....	162
Add directional association.....	164
Add compensation association.....	165
Alignment tools.....	165
Alignment actions.....	166
Align top.....	166

Align bottom.....	166
Align middle.....	166
Align left.....	166
Align right.....	166
Align center.....	166
Nudge tools.....	166
Nudge right.....	167
Nudge left.....	167
Nudge up.....	167
Nudge down.....	167
The sizing tools.....	167
Make same height.....	167
Make same width.....	168
Make same size.....	168
The Space equalization tools.....	168
Equalize horizontal spaces.....	169
Equalize vertical spaces.....	169
Canvas level context menus.....	171
Grid.....	171
Grid properties.....	171
Rulers.....	171
View rulers.....	172
Ruler Tracking.....	172
Zoom.....	172
Zoom In.....	172

Zoom Out.....	172
Custom Zoom.....	172
Zoom to Fit.....	173
Zoom to Selection.....	173
Diagram Color.....	173
Measurements and Size.....	174
How to increase the size of your process map.....	174
Default properties.....	177
Line properties.....	177
Font selection.....	177
Fill options.....	177
Preferences Settings.....	178
Graphical Objects Label Orientation.....	179
Connecting Objects Label Orientation.....	179
Percentage along line.....	179
Distance from line.....	179
Show Caption for:.....	179
Process Navigator Actions.....	181
Common Context Menu Actions.....	181
Copy.....	182
Cut.....	182
Paste.....	182
Delete.....	182
New.....	182
Project.....	183

Add.....	183
Renaming items.....	184
Task object actions.....	185
New Property.....	186
Join Message flow.....	187
Join Sequence flow.....	187
Join Association.....	188
Join Compensation association.....	188
Join Exception flow.....	188
Diagram object actions.....	189
New Pool.....	189
New Property.....	189
New Data object Artifact.....	190
New Group artifact.....	190
New Annotation artifact.....	190
SubProcess object actions.....	192
Independent SubProcess.....	192
Event object actions.....	194
New Property.....	194
New Message Trigger.....	194
New Timer Trigger.....	195
New Exception Trigger.....	195
New Cancel Trigger.....	195
New Compensation Trigger.....	195
New Rule Trigger.....	195

New Link Trigger.....	195
New Terminate Trigger.....	195
Join Message flow.....	196
Join Sequence flow.....	196
Join Association.....	196
Join Compensation Association.....	196
Join Exception Flow.....	198
Process object actions.....	199
New SubProcess.....	199
New Task.....	199
New Property.....	199
New MetricsSet.....	199
New XOR Gateway.....	200
New OR Gateway.....	200
New Complex Gateway.....	200
New AND Gateway.....	200
New Start event.....	200
New Intermediate Event.....	200
New End Event.....	201
New Data object artifact.....	201
New Group.....	201
New Annotation.....	201
Connecting objects.....	203
Two disconnected Task objects to be joined.....	204
Creating a Message flow connection between two Task objects.....	209

Example: Creating an Association.....	213
The create association action.....	215
Properties.....	218
How Avantage implements properties.....	218
Standard Properties.....	218
Changing the type of a property.....	236
Editing property values.....	221
String (text) properties.....	222
Other properties.....	223
Measure points and metrics.....	224
The Log Viewer .....	226
The Platform.log file.....	227
Starting the Log Viewer.....	228
Log Viewer Filters.....	229
Organization Modeling.....	231
The organizational structure.....	231
Adding organization files to a project.....	232
To add a new file.....	232
To add an existing file.....	233
Working with organization files.....	234
Enabling an organization file.....	235
Adding departments.....	236
Adding roles.....	236
Deleting organization units.....	236
Linking BPMN pools and swimlanes to the organization structure.....	238



Unlinking.....	239
Working with sections.....	240
What is a section?.....	240
Exporting a section.....	240
Importing a section.....	240
Section placement rules.....	240
Appendix A: Process Model Settings.....	241
Main settings.....	241
Database connection chooser .....	242
Forcing a settings change to persist immediately.....	242
Metrics Settings.....	242
Appendix B: File types.....	243
File types in projects .....	243
Appendix C: BPMN Complete set.....	244
BPMN Complete Set.....	244
Index.....	246



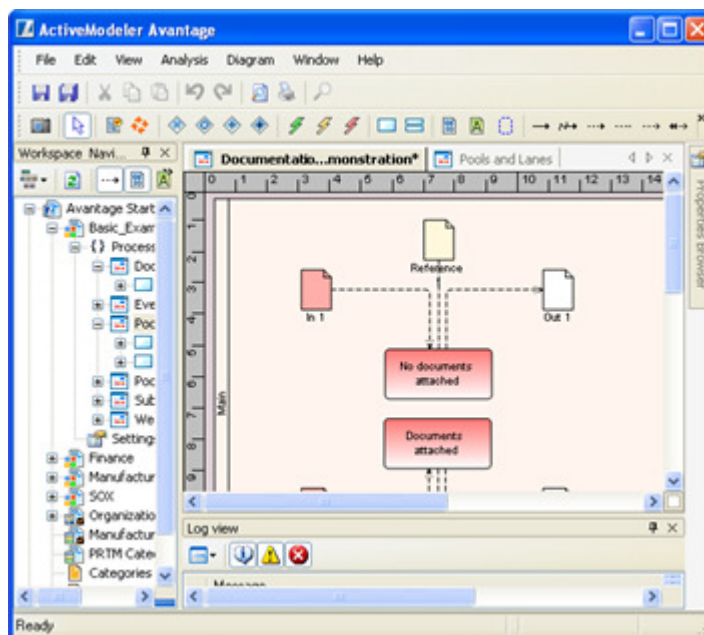
# Introducing the ActiveModeler Avantage BPM Platform

*Avantage is a new generation standards oriented BPM (Business Process Modeling) platform. The ActiveModeler Avantage Platform provides BPMN 1.0 compliant process diagram editing, multiple projects, and a .NET process modeling object model (framework).*

## About ActiveModeler Avantage

ActiveModeler Avantage is a new generation BPM tool that was designed specifically with extensibility in mind.

Avantage was specifically designed to be extensible. You can add new and updated features simply by downloading a plugin.

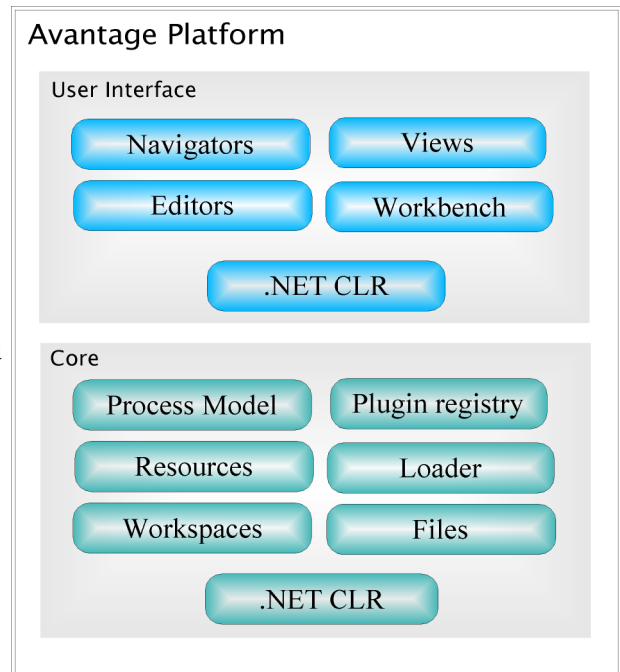


## Features and benefits

Feature	Benefit
<ul style="list-style-type: none"> <li>▪ <b>BPMN Diagramming</b></li> </ul>	<p>You can create BPMN 1.0 compliant process diagrams. The data is stored in an XML based document.</p>
<ul style="list-style-type: none"> <li>▪ <b>Project oriented</b></li> </ul>	<p>Avantage lets you organize your work into projects.</p>
<ul style="list-style-type: none"> <li>▪ <b>Plugin based</b></li> </ul>	<p>Features and functionality can be added to the basic Avantage Platform software by downloading additional plugins.</p>
<ul style="list-style-type: none"> <li>▪ <b>Process Navigator</b></li> </ul>	<p>An hierarchical process model editor/browser makes it easy to manage both small and large projects and multiple projects.</p>
<ul style="list-style-type: none"> <li>▪ <b>.NET BPMN Process Object Model</b></li> </ul>	<p>You can use common languages like C#, VB.NET Delphi and others to extend functionality. There is a deep process modeling API, making it easy to develop advanced process modeling software on the Avantage platform. Can script the APIs with Lua.</p>
<ul style="list-style-type: none"> <li>▪ <b>Lua .NET scripting engine</b></li> </ul>	<p>Avantage includes a Lua plugin. Lua scripts run as .NET CLI and have full access to .NET, Avantage and the BPMN process model and other plugins.</p>
<ul style="list-style-type: none"> <li>▪ <b>Category modeling</b></li> </ul>	<p>Define categories which can then be associated to objects in a process model. Includes a category editor.</p> <p>Dynamically define process model entity labels and colors depending on the category information it is associated with.</p>
<ul style="list-style-type: none"> <li>▪ <b>Organization modeling</b></li> </ul>	<p>Avantage provides deep organization modeling. This where you define the company structure within which the processes are deployed. The structure is composed of Departments (these are hierarchical) and Roles (people and machines).</p> <p>You can map BPMN container entities like <b>Pools</b> and <b>Swimlanes</b> to a node in the organization model.</p>
<ul style="list-style-type: none"> <li>▪ <b>XPDL 2.0</b></li> </ul>	<p>The XPDL 2.0 plugin provides XPDL 2.0 import and export of BPMN process models. Category and Metrics attributes from the Avantage Analytics plugins are output if available.</p>
<ul style="list-style-type: none"> <li>▪ <b>Metrics</b></li> </ul>	<p>The Metrics plugin instruments your BPMN process model with powerful numerical and cost analytics.</p> <p>Calculate the financial costs of a process. Make improvements and cost savings. Create Excel reports.</p>

# Quick Introduction to the architecture of ActiveModeler Advantage

- Process modeling library built-in.
- BPMN object model.
- BPMN diagram editor.
- BPMN process model navigator
- Completely extensible. You can extend any part of the platform, User Interface or Core, i.e. a View, a Navigator, a BP object, a Menu, a Toolbar, or even the BPMN object model itself.



# Installation

*How to install and setup Avantage on your computer*

## Prerequisites

- ▶ .NET Framework 2.0 or 3.0
- ▶ Windows 2000, XP or later
- ▶ Intel Pentium 4 class or later compatible processor.
- ▶ Minimum of 1GB RAM.

## Installing the .NET Framework 3.0

[Microsoft .NET Framework 3.0 Redistributable Package](#)

## How to determine which version of the .NET Framework is installed on your computer

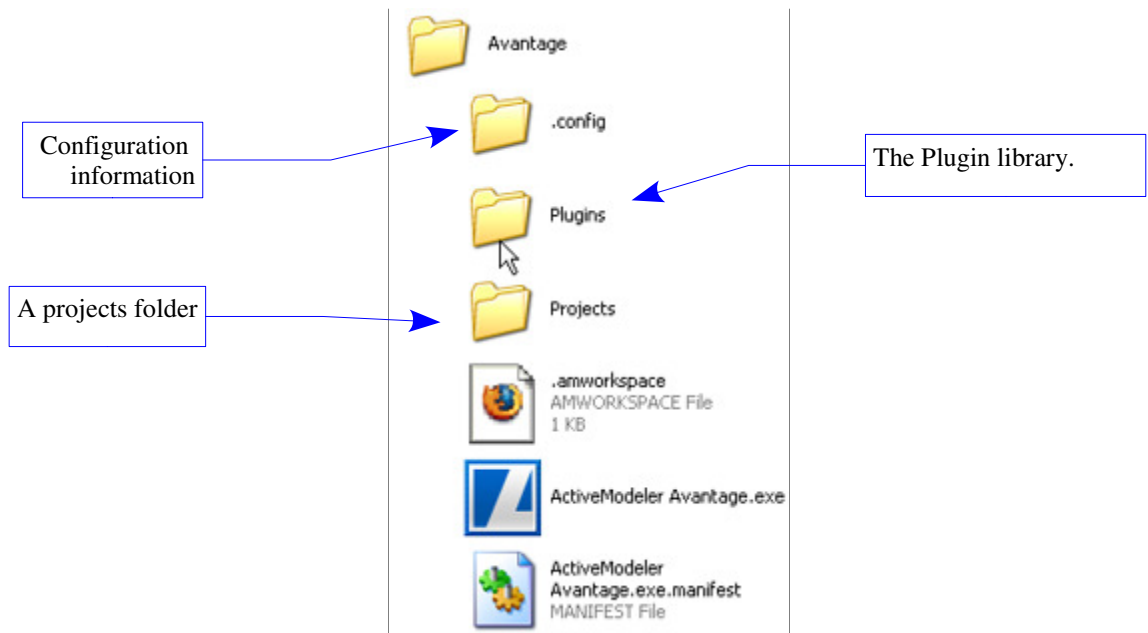
[Microsoft knowlegebase article](#)

## Running Setup

To install the Avantage Platform on your computer run the “Avantage Platform Setup.exe” file.

The installation process is straightforward and it makes an ultra-lightweight footprint in your system since no COM components are deployed.

**Figure 1.** The installation structure



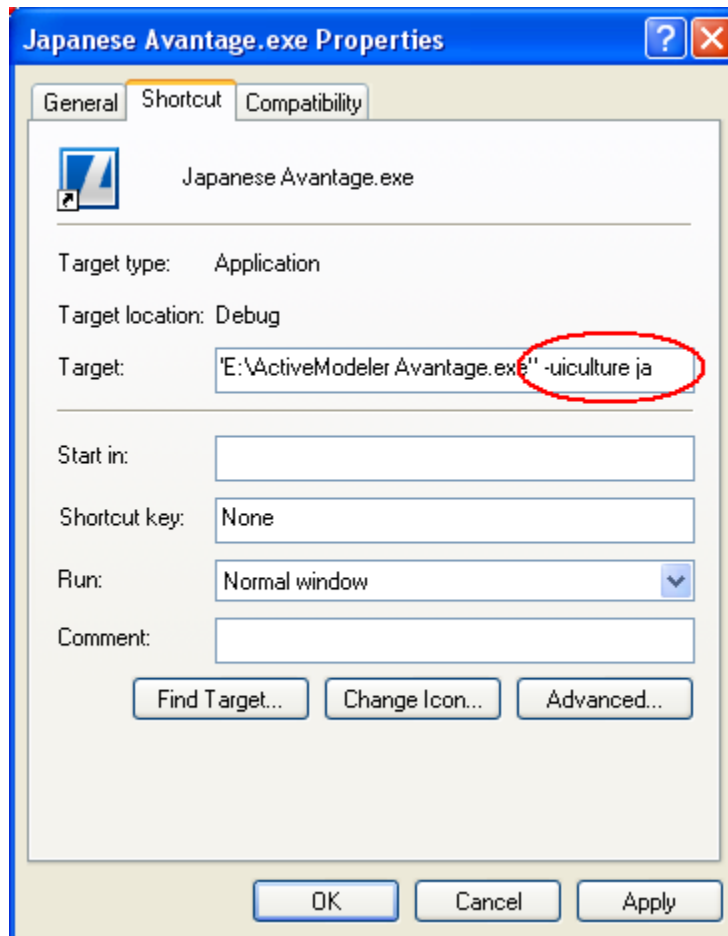
## Multi lingual deployment

ActiveModeler Avantage supports multi-lingual deployment. The command line argument **-uiCulture** allows you to specify the language at start up.

An easy way to supply command line arguments is to make file shortcut to the Avantage.exe and put the necessary parameters into the shortcut. In the example below the shortcut was named “Japanese Avantage”.

*Note:* The Avantage Platform Setup process creates both English and Japanese shortcuts in the Start menu.

**Figure 2.** Setting the **-uiCulture** with the shortcut properties dialog

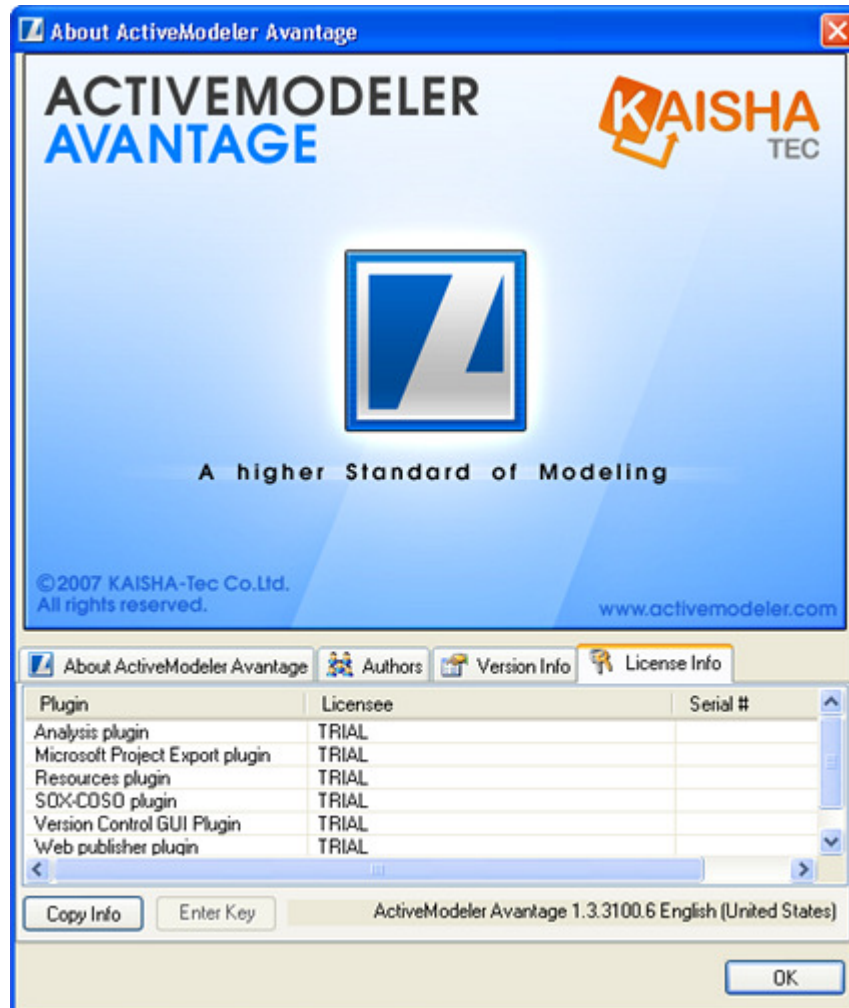


Check the KAISHA-Tec website for the latest list of supported cultures..

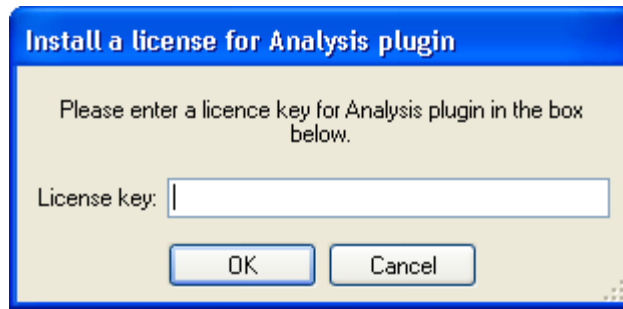


## Adding plugin licenses

When you purchase a commercial plugin you will receive a license key. To add a license, bring up the Avantage about dialog – do this by selecting the **Help>About ActiveModeler Avantage...** menu, then click on the **License Info** tab to display the license information page.



To add a license for a plugin, select it first in the list (this will enable the **Enter Key** button). Click the Enter Key button, this will display the license dialog.



Enter the license key for the plugin. Please note that the license will not be recognised until you restart Avantage.

# An introduction to BPMN

*This chapter introduces you to the Business Process Modeling Notation (BPMN) and explains the graphics that make up the notation to form a Business Process Diagram (BPD).*

BPMN allows you to work very simply at a high level or in detail for a full “in-depth” analysis, the level of precision in the diagram depending on the result you want to achieve. A simple diagram can also be enhanced later for a more complete analysis of course.

BPMN is a core feature of Business Process Management (BPM) and we will look at some of the concepts and standards involved in this area. BPM is in fact bringing together the previously separate areas of Process Modeling, Simulation, Workflow, Enterprise Application Integration (EAI), and Business-to-Business (B2B) integration into a single standard.

## Why BPMN?

Historically, each vendor provided their own individual notations in their tools and methodologies. This was also the case for ActiveModeler. Our first product offering was introduced in 1995, before the existence of any well-defined standards.

The Business Process Management Initiative (BPMI) brought together a consortium of process modeling vendors and consultants to work on a common standard notation and importantly the business user was the target for the standard. Their website is at [\[\[http://www.BPMI.org\]\[BPMI.org\]\]](http://www.BPMI.org). The BPMI team consolidated the best ideas from many of the divergent notations on the market into a single notation.

BPMN also includes a key capability: an ability to produce a mapping from the business-oriented notation to an execution language e.g. BPEL4WS. This

creates a new standardized bridge between the business analysis and the IT implementation that was previously handled by ad-hoc methods.

The end users of standardized process modeling tools will benefit greatly because the models they develop will be consistent regardless of the tool of origin or purpose. This will greatly aid communication, collaboration and standardization.

The mapping to execution languages provides a bridge that was before handled by IT developers re-creating process models for input into the appropriate execution tool. Such a manual connection between the business-level model and the IT-level model is error prone and loses traceability between the original requirements and the executed processes. So the possibilities for BPMN look very exciting.

In adopting BPMN as a notation, KAISHA-Tec is bringing its many years of process modeling knowledge into the BPMN arena and allowing both customers and our organization itself to benefit from the opportunities of a standards based platform.

## What Is BPMN?

The Business Process Modeling Notation (BPMN) is a new standard to model business process flows and web services and the first official release to the public was made in May 2004. The notation specification represents more than two years of effort by the BPMI Notation Working Group.

The primary goal of the BPMN standard was to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology and finally, to the business people who will manage and monitor the processes.

An equally important goal of BPMN was to provide an internal model (often referred to as an **object model** or **process model**) that will enable the generation of XML languages for the execution of business processes e.g. BPEL4WS (Business Process Execution Language for Web Services) and BPML (Business Processing Modeling Language). Therefore, BPMN creates an important standard bridge between the business process design and process implementation.

BPMN was developed with a solid mathematical foundation so that a precise execution language can be produced.

The end result of BPMN is a Business Process Diagram (BPD), which shows the flow of activities based on a standard graphical notation.

A business process diagram is not the only result however. By defining the process diagram a BPMN process model has also been created by the modeler. Many modelers lock this information away and only provide access to it through something like a BPEL4WS export or Save as... mechanism. ActiveModeler Avantage is specially designed to *publish* and make accessible the underlying process model so that it can be accessed directly from plugin extensions.

## BPMN Basics

The Business Process Diagram (BPD) is designed to:

Be easily understandable by non-technical people (remember the target was the business user)

Have the eloquence to be able to model even the most complicated processes

Offer a natural mapping to business execution languages

The BPD is made up of a set of graphical elements. The four basic categories of elements are:


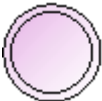
1. Flow objects
2. Connecting objects
3. Swimlanes
4. Artifacts



## Flow Objects

BPMN has 3 three flow objects as follows:


### (i) Event

An Event is represented by a circle and is something that “happens” during the course of a business process. There are 3 types of events:

	<b>StartEvent</b>	This kind of event starts a process flow
	<b>Intermediate Event</b>	This kind of event happens during the course of a process flow

	<b>StartEvent</b>	This kind of event starts a process flow
	<b>EndEvent</b>	This kind of event ends a process flow

Events affect the flow of the process and usually have a cause (trigger) or an impact (result). Events have open centers and can optionally show different triggers or results in them. There is a standard set of these informative events and we will cover them later.

	For example the symbol on the left is a message start event which means that a message arrival initiates a
---	--

**(ii) Activity**

An Activity is represented by a rounded rectangle and is a generic term for a work item. An Activity can be either Atomic – i.e. not further decomposable, in which case it is called a Task or it can be non-Atomic i.e. it is further decomposable, in which case it is called a Sub-Process. The Sub-Process is distinguished by a small plus sign in the bottom center of the shape.



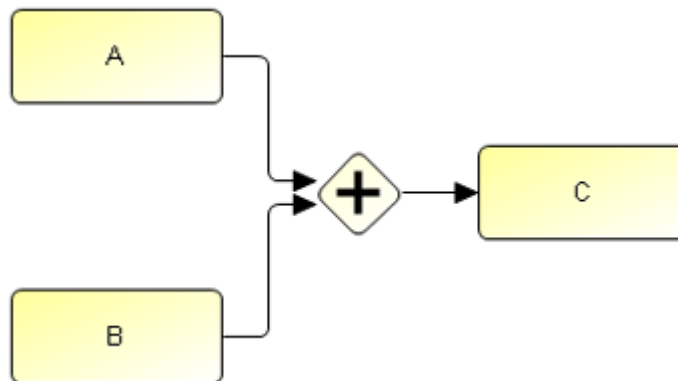
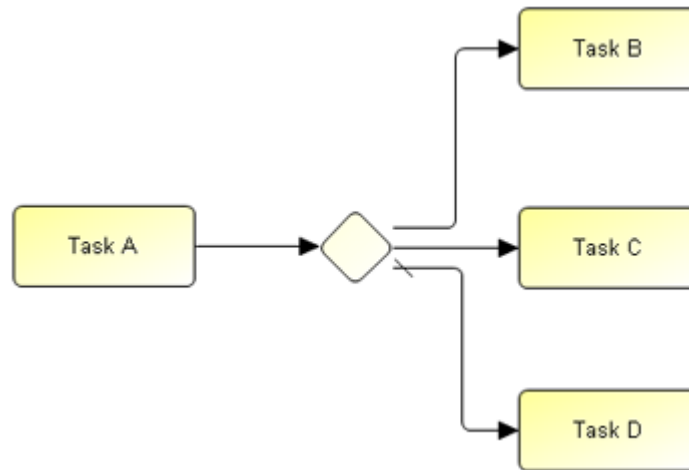
Task 1

➔ **NOTE:** You will often see a mention of a BPMN **Process** too. A Process is is not a specific graphical object, rather it is a set (or collection) of graphical objects.

**(iii) Gateway**

A Gateway is represented by a diamond shape and is used to control the divergence and convergence of the process flow (splits and joins).

A sign in the diamond itself can show the type of gateway. In the diagram below a “plus sign” is used to denote an AND-Join or synchronization.



## Connecting Objects

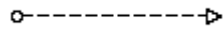
Flow Objects are connected together by Connecting Objects to create the basic structure of a business process. There are three types of Connecting Objects:

### (i) Sequence Flow



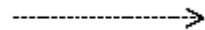
A Sequence Flow is represented by a solid line with a solid arrowhead and is used to show the activity sequence order.

**(ii) Message Flow**



A Message Flow is represented by a dashed line with an open arrowhead and is used to show the flow of messages between two separate Process Participants (business entities or business roles) that send and receive them. We will see a little later that in BPMN, two separate Pools in a diagram will represent two Participants.

**(iii) Association**



An Association is represented by a dotted line with a line arrowhead and is used to associate data, text, and other Artifacts with flow objects. Associations are used to show the inputs and outputs of activities.





### Allowable Sequence flow connections

An incoming sequence flow can connect to any location on a flow object (top, left, bottom, right). Similarly, an outgoing sequence flow can connect from any location on a flow object (left, right, top, or bottom). Message flow also have this capability.

**Figure 3.** Allowable sequence flow connections



From/To						
						
						
						
						
						
						

## Allowable Message flow connections

The table below shows the BPMN modeling objects and shows how these objects can connect to one another through message flow. The message flow symbol indicates that the object listed in the row can connect to the object listed in the column.

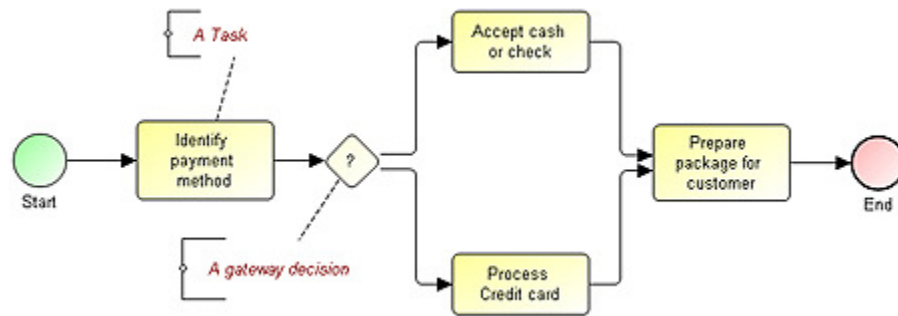
**NOTE:** that message flow cannot connect to objects that are within the same *Participant Lane* boundary.

**Figure 4.** Allowable Message flow connections

From/To						
						
						
						
						
						
						

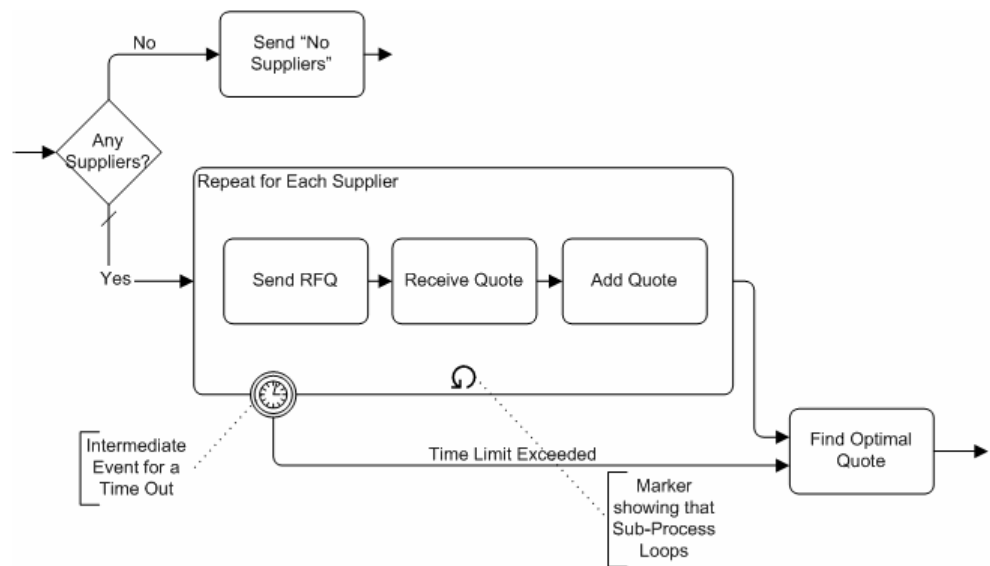
If a process analyst only requires a process diagram with a low level of precision, the core elements plus the connectors will provide the ability to easily create understandable diagrams to convey process flows. Such a diagram is shown below:

**Figure 5.** A low precision diagram:



For process analysts who require a higher level of precision to create process models, which will be subject to detailed analysis or will be managed by Business Process Management System (BPMS), additional details can be added to the core elements and shown through internal markers.

An example is shown below:

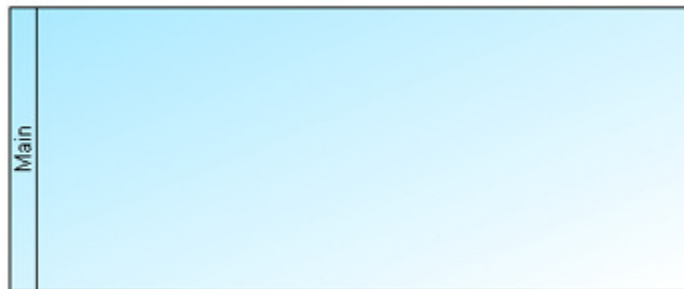


## Swimlanes

Swimlanes are used to group activities into distinct visual categories to highlight different functional capabilities or responsibilities. BPMN supports swimlanes with two main constructs:

### (i) Pool

A Pool is physically separated in a BPD and represents a Participant in a Process. It is also acts as a graphical container for partitioning a set of activities from other Pools.



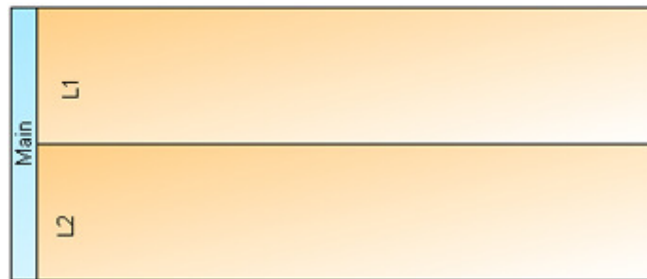
There is always at least one pool in a BPD diagram, it is a normally borderless (invisible) pool called Main. Additional Pools are used when the diagram involves two separate business entities or participants. The activities within separate Pools are considered self-contained Processes and the Sequence Flow may not cross the boundary of a Pool.

Message Flow is defined as being the mechanism to show the communication between two participants, and, thus, must connect between two Pools (or the objects within the Pools).

Often pools are used for modeling of B2B processes.

### (ii) Lane

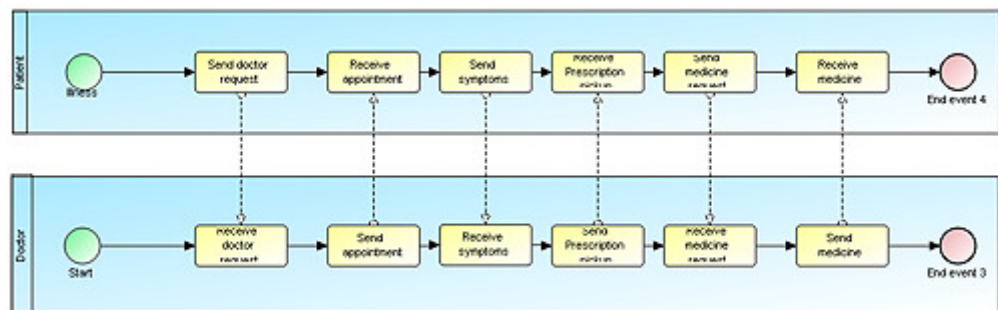
A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities.



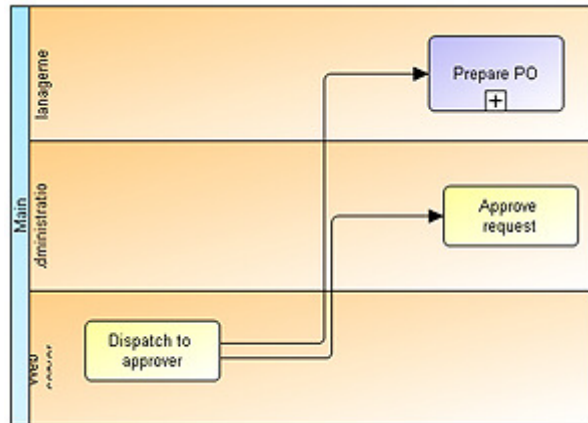
Lanes are similar to the traditional swimlane process modeling methodologies. They were used also in ActiveModeler 2.5. Lanes are often used to separate activities associated with a specific company function, department or role and Sequence Flow may cross the boundaries of Lanes within a Pool. Importantly, Message Flow may not be used between Flow Objects in Lanes of the same Pool.

We can see an example of a BPD with pools below:

**Figure 6.** BPD with pools



**Figure 7.** Process segment with lanes:

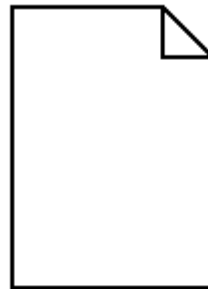


## Artifacts

BPMN was designed to allow some flexibility in extending the basic notation and in providing the ability to additional content appropriate to a specific modeling situation, such as for a vertical market (e.g., insurance or banking). Any number of Artifacts can be added to a diagram as appropriate for the context of the business processes being modeled. The current BPMN specification defines three types of BPD Artifacts, which are:

### (i) Data Object

Data Objects are a mechanism to show how data is required or produced by activities. They are connected to activities through Associations.



Purchase Form [Running]

### (ii) Group

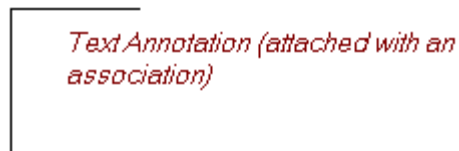
A Group is represented by a rounded corner rectangle drawn with a dashed line. The grouping can be used for documentation or analysis purposes, but does not affect the Sequence Flow.



Group Artifact

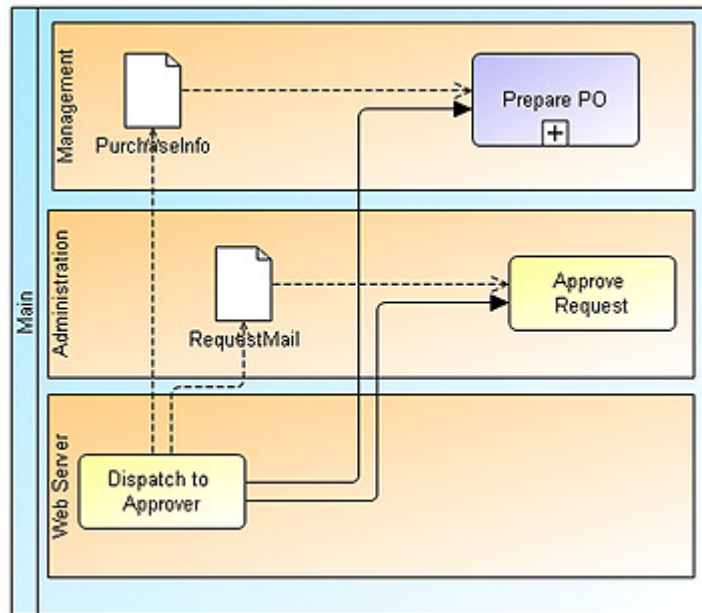
(iii) Annotation

Annotations are a way for process analysts to provide additional text information to the reader of a BPMN Diagram.



Annotation 1

The BPMN standard allows for process modeling software vendors to define additional kinds of artifacts which add more details about how the process is performed. However, the basic structure of the process, as determined by the Activities, Gateways, and Sequence Flow, are not changed with the addition of Artifacts in the diagram





## General uses of BPMN

We have covered the basic features of BPMN in this chapter. Now we show how to use this modeling methodology. There are two basic types of models that can be created with a BPD:

- Internal Business Processes
- Collaborative B2B Processes

### Internal business processes

An internal business process will generally be focused on a single business organization.

Although internal processes often show interactions with external participants, they define activities that are not generally visible outside the organization (i.e. not in the public domain and are still therefore regarded as private activities. If swimlanes are used then an internal business process will be contained within a single Pool. The Sequence Flow of the Process is therefore contained within the Pool and cannot cross the boundaries of the Pool. Message Flow can cross the Pool boundary to show the interactions that exist between separate internal business processes. In this way a single Business Process Diagram may show multiple private business processes.

### Collaborative B2B Processes

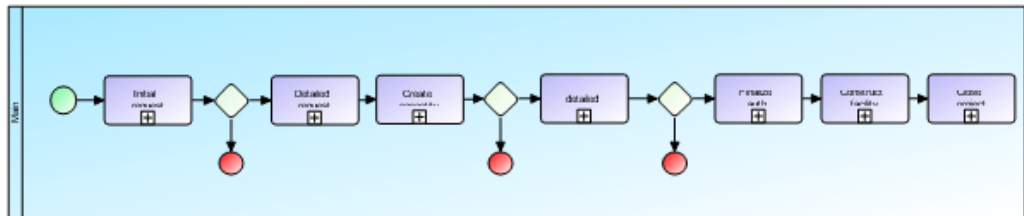
A collaborative B2B process shows the interactions between two or more business entities. The diagrams for these types of processes are generally from a global point of view. That is, they do not take the view of any particular participant, but show the interactions between the participants. The interactions are shown as a sequence of activities and the message exchange patterns between the participants. The activities for the collaboration participants can be considered the “touch-points” or “contact points” between the participants. When looking at the process shown in only one Pool (i.e., for one participant), public process is also called an abstract process. The actual (internal) processes are likely to have more activities and detail than what is shown in the collaborative B2B processes.

An example of a collaborative B2B process is shown below:

## Different levels of precision

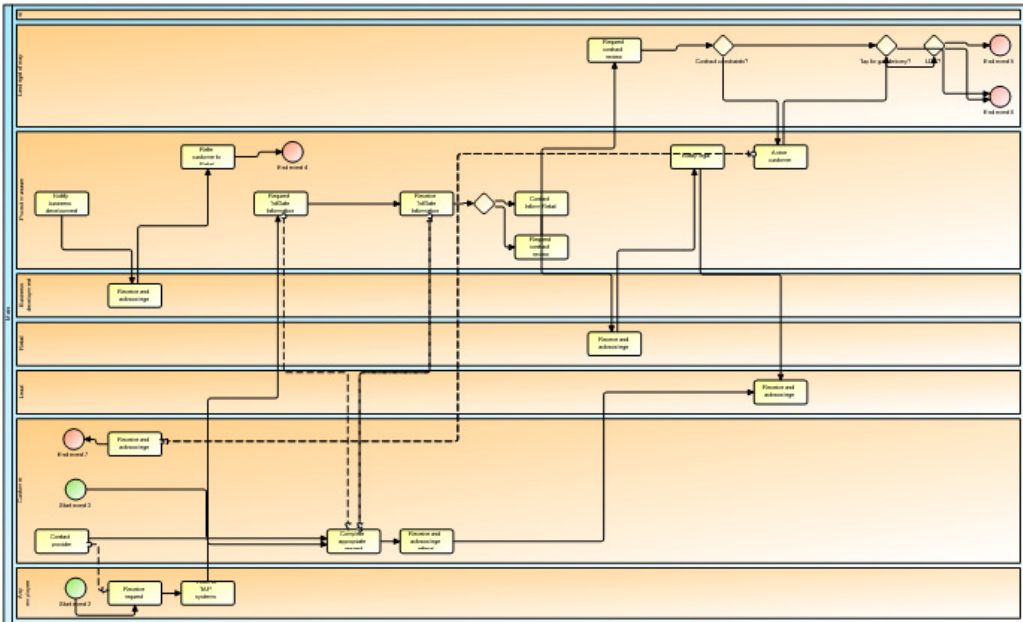
The modeling of business processes often starts with capturing high-level activities and then drilling down to lower levels of detail within separate diagrams. BPMN allows you to start simply and add detail in the form of Sub-Processes as necessary to add to the knowledge capture.

We see below an example of a high level process, which is basically a series of Sub-Processes with three decision points in the Process.



Next we move on to see the details of one of the Sub-Processes. This diagram employs two (2) Pools. One is for the customer and one for the company providing the service. Note that this diagram shows the internal business process for the company and shows an abstract process for the customer (i.e., the customer process only includes the activities used for communicating through Message Flow to the company). The activities within the company are partitioned by Lanes to show the departments or roles responsible for their performance (e.g., System Coordinator, Business Development, Legal, and Retail).

Figure 8. Details of a subprocess



## Mapping a BPMN diagram to Execution Languages

A key goal with BPMN was to create a bridge from the business-oriented process modeling notation to IT-oriented execution languages. The graphical objects of BPMN, supported by a rich set of object attributes, are readily mappable to process execution languages such as BPEL4WS (Business Process Execution Language for Web Services) which is becoming a defacto standard for process execution.

ActiveModeler Avantage will provide bridges to various execution languages and workflow engines through its unique pluggable architecture as new standards emerge. The aim here it not to be locked in to one standard, but to host any standard.

### XPDL 2.0 Plugin

The XPDL 2.0 Plugin provides support for importing and exporting BPMN process models.

## The Road ahead

It seems very likely that many companies will support the BPMN standard and further integration to execution languages and program development environments e.g. UML will take place.

For BPMN itself, the plan is to make maintenance versions available, perhaps annually. The goal is also to produce standardized Artifact sets to support vertical business markets e.g. insurance, manufacturing, and finance. In addition, BPMN will be a good platform for higher-level business modeling which includes, for example, the modeling of business rules and business strategy.

BPMN has now been formally adopted as a standard by the Object Management Group (OMG). Considering that the OMG is currently developing UML 2.0, which includes Activity Diagrams, it is possible that a consolidation of BPMN Business Process Diagrams and UML Activity Diagrams will take place in the future.

# Concepts and terminology

*This chapter explains the basic “Must Know” concepts and terms in order for you to get the most value from using Avantage.*

## Workspaces

An Avantage Workspace is a combination between the start page and a solution in Visual Studio.

At any one time, only one workspace is available in Avantage. The workspace contains the available projects and other information like the views open (current perspective) when the workspace was last saved. There is one workspace per user. It is automatically loaded when Avantage starts and saved automatically when Avantage is closed.

The workspace contains a list of projects. A project can be added to more than one workspace (different users on the same or different machines). In the workspace tree, the projects are root nodes.

The workspace is an xml file on disk, located in the root folder of Avantage's installation. The workspace files are named after the users to whom they belong.

The workspace contains only a list of projects previously opened by the user and left in the workspace.

The contents of your Workspace are displayed in the Workspace Navigator (which is displayed as a tree). You can add and remove entries into your workspace, create folders, open and edit information through Actions (commands).

Each kind of entry in the Workspace tree has a set of actions assigned to it. You can see the commands assigned to any item by right-clicking on it.

## Projects

A project consists of a list of files and folders located in the project's directory or a subdirectory. Only a subset of files and sub-folders from the project's directory are included in the project folder and they must be specifically added to it (through commands in Avantage).

No files outside the project's folder are available for inclusion in a project.

There is only one project per root folder. The name of the file is **".project"**. When adding a project to the workspace, if there is no such file but there are other files with extension **.project** in a folder, Avantage will ask which file to add.

Besides files, a folder can contain other information like currency used. This information is saved in the project file in xml format.

## Files

A file is the unit that can be edited. Special files like process model files are expandable to show other entities that can be edited (and which 'live' inside the file) but the file will be the one to go through editing lifetime.

If there is an editor registered for a file extension (i.e there is a suitable plugin installed), it will get activated when the file is double-clicked in the navigator tree. If there's no editor registered for a file, double-clicking on it will do nothing.

## Views

A view is a workspace component that can navigate a hierarchy of information or display properties for an object. Only one instance of any given view is open in the workspace at any time. When the user makes selections or other changes in a view, those changes are immediately reflected in the workspace. Views are often provided to support a corresponding editor.

A view has the following attributes and behaviour:

- ▶ Has Dynamic pop-ups, in addition to static ones
- ▶ Has an immediate action - changes are saved immediately
- ▶ Only one instance of view exists at a time. You cannot have more than one window showing the contents of the same file.

- ▶ Is selection aware (the workspace will take care of notifying the interested listeners)

## Editors

An editor is the workspace component that allows you to edit an object (often a file). Editors operate in a open/save/close life cycle. Changes made in an editor are not committed until you save them.

An editor has the following attributes and behaviours:

- ▶ Open/Save/Close lifecycle
- ▶ Has a Toolbar and menu - (a single toolbar/menu for all editors of same type)
- ▶ Has Dynamic pop-ups, in addition to static ones
- ▶ Is Selection aware (the workspace will take care of notifying the interested listeners)
- ▶ Undo/Redo support
- ▶ Many instances of the same editor type can run at a given time.

## Actions (commands)

Workspace components can have actions which are chosen from toolbar icons, menus and dynamic pop-ups depending on the type of workspace component.

## Plugins

A plugin is a .NET .dll file that advertises to the platform what functionality it provides. Plugins are used to add features to and extend the Advantage BPM Foundation application.

You are probably familiar with the phenomenon of "code bloat". This happens because many software manufacturers try to package as many conceivable features as possible into a product in the hope that everyone will be happy. Most people use very few features and the unused features are typically expressed as wasted disk space and increased fragility in the product.

Plugins contain specific "chunks" of functionality (for example a report, an editor, a navigator, a BP model, a filter). You only install the plugins that you need for your particular requirements, saving memory and disk space.



Avantage is supplied with a default set of plugins to enable you produce BPMN diagrams and process models.

If you are a software developer or ISV and you would like to produce plugins for Avantage then you will need our Developer's Kit. Please contact KAISHA-Tec for details of the Developers Kit.

## **Simple configuration**

Avantage uses xml format text files for all configuration. Your installation includes a default configuration.

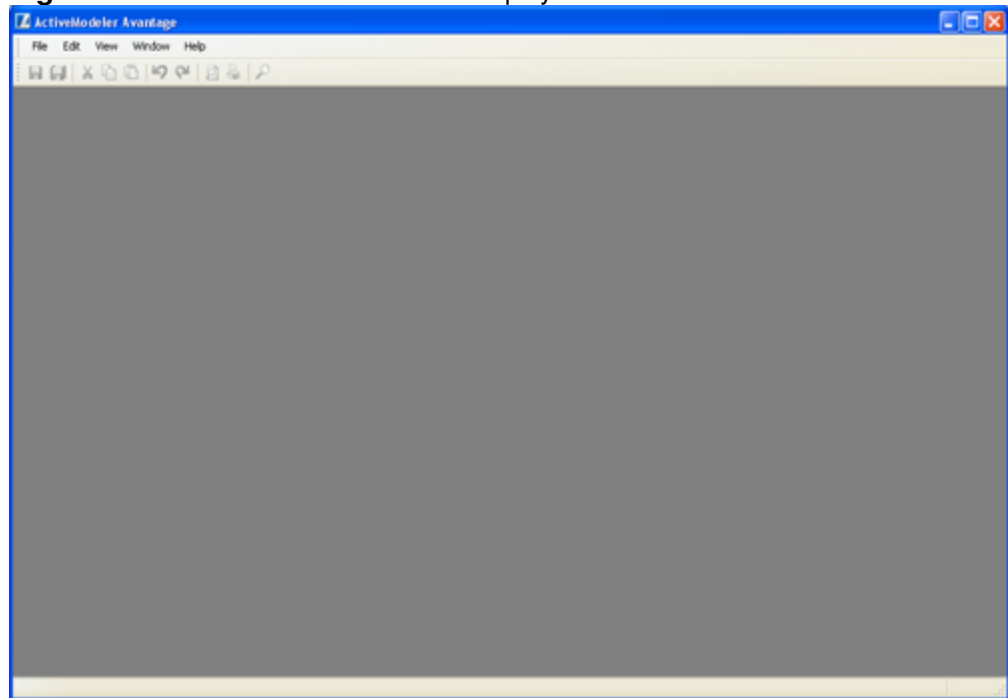
# QuickStart Guide

*This chapter will help you get started using Avantage as quickly as possible.*

## The first time you run Avantage

When you start Avantage for the first time there isn't very much to see. You see the default *Workbench*. In order to make the application more useful we need to display the Workspace Navigator and other views. These are available from the *View* menu.

**Figure 9.** The *Workbench* window is displayed the first time

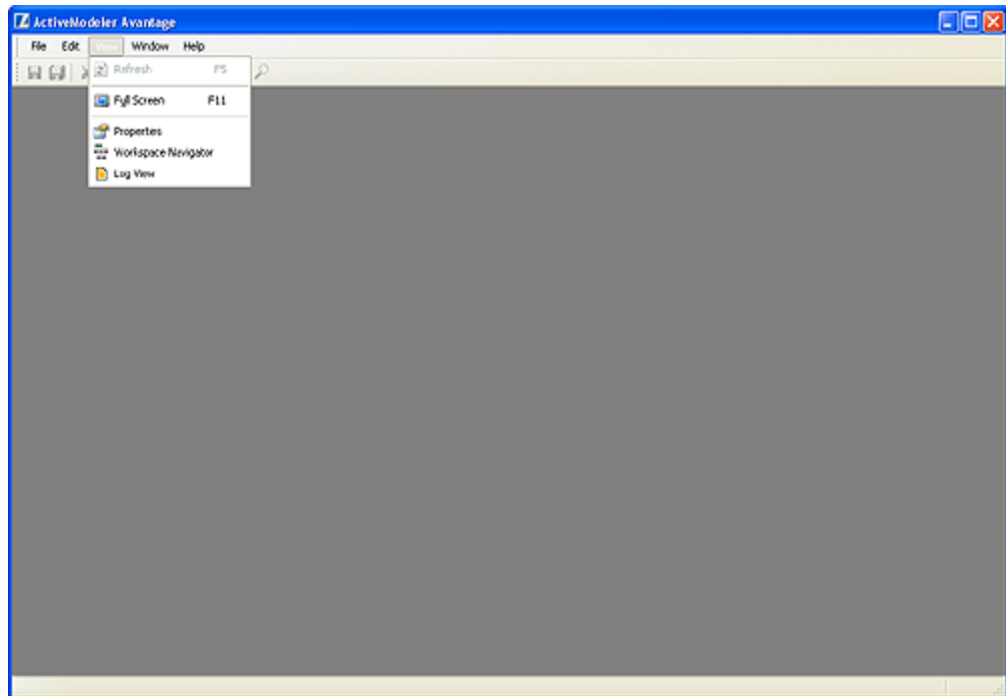


➔ It is necessary to choose and arrange which views you will work with. Avantage will then remember your settings for next time.

Choose the *View:Workspace Navigator* menu from the menubar.

**Figure 10.** The View Menu

*Note: What you see in the View menu depends on what kind of View plugins are installed. The Workspace Navigator plugin is **always** installed.*

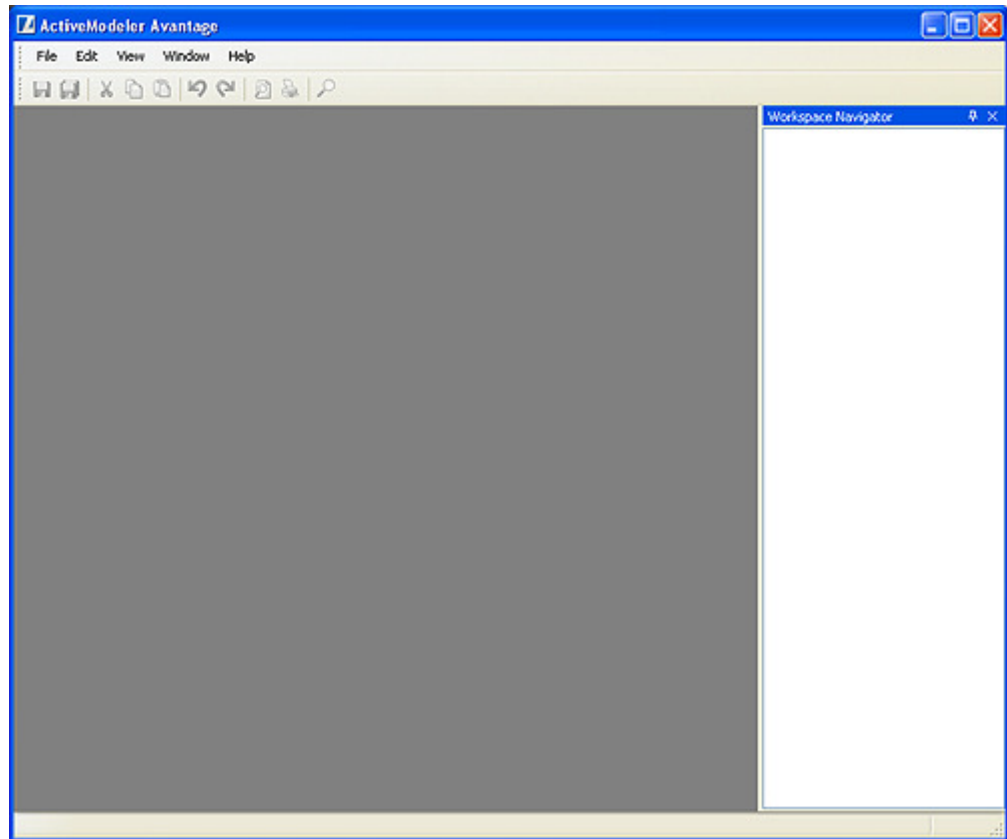


What you see in the View menu depends on which plugins are installed. You should at least see the following three menu items regardless of which plugins you have.

- ▶ Workspace Navigator
- ▶ Properties Browser
- ▶ Log View

Choose the Workspace Navigator menu. This will make the Workspace Navigator visible. By default it is displayed docked on the right hand edge of the main window, like in the Figure below:

**Figure 11.** The Default Workspace Navigator View



You can relocate (dock) the workspace window to any of the edges of the main workbench window. Relocating a view window is easily done by dragging the title bar of the Workspace Navigator to the edge of the Workbench window you want to dock with.

Some people prefer to have the Workspace Navigator view on the left hand side of the Workbench window. Let's move it there now.

Drag the Workspace Navigator window by its title to the left hand side of the workbench, release the mouse on the left hand edge of workbench window.

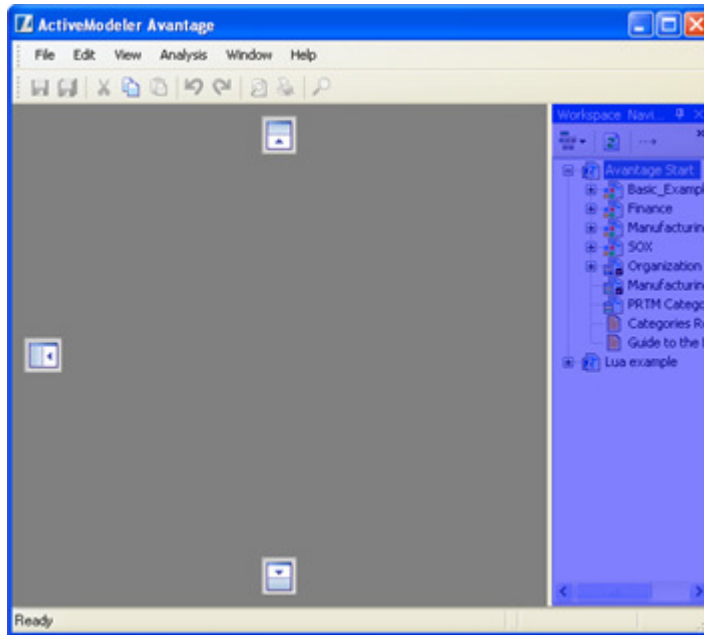
➔ *Note:* If the Navigator floated instead of docked you did not release the mouse on the edge of the workbench window. You will have to try it again.

## Working with docking windows

Avantage includes a very powerful docking manager which enables you to make the best use of the limited screen real estate of your monitor. It is a good idea to familiarize yourself with it.

### To move and re-dock a docked window

**Figure 12.** Step 1: Click on the window's title bar.



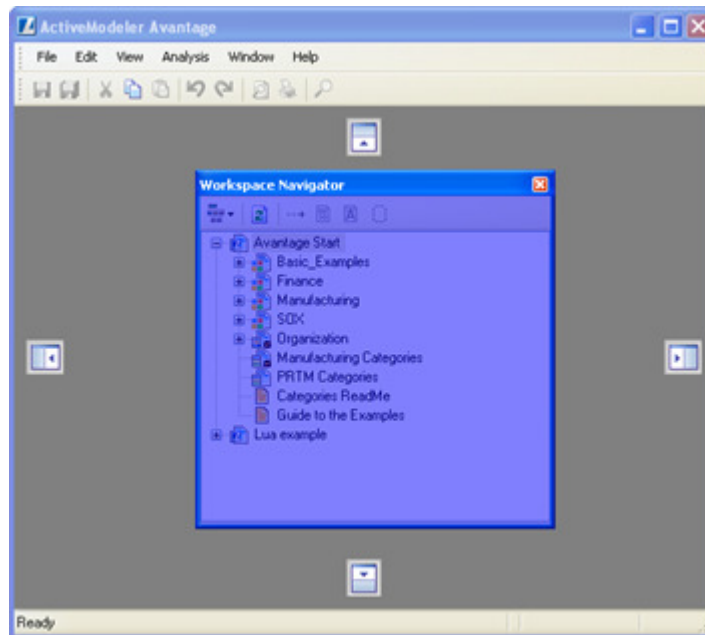
While the mouse is held down, slowly begin to drag the mouse. Notice that the docked window becomes highlighted with a mauve tint. This indicates that the window you have clicked can be re-docked.

As well as the highlight colour you will also see the docking ports appear. These ports are the targets to which you can move the window. The top port allows you to dock to the top edge, the left port allows you to dock to the left edge and the bottom port allows you to dock to the bottom edge and so on.

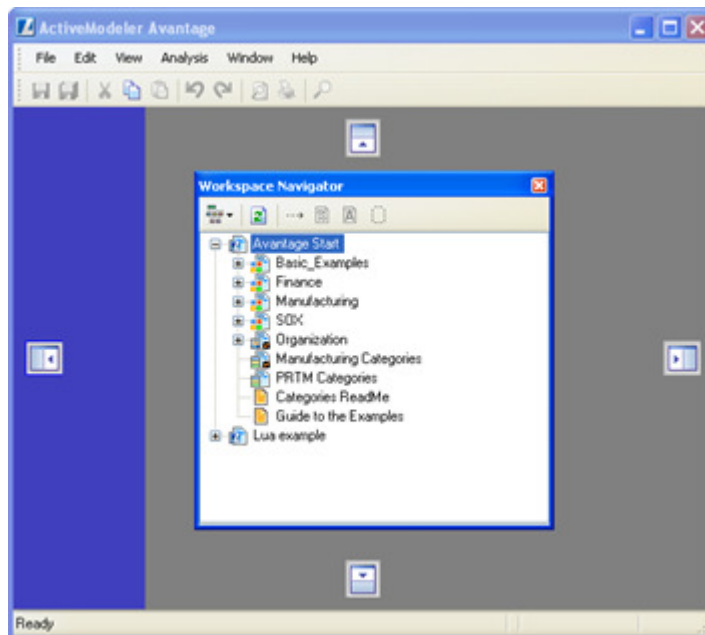
**Figure 13.** The docking ports



**Figure 14.** There is a docking port for each side of the main window

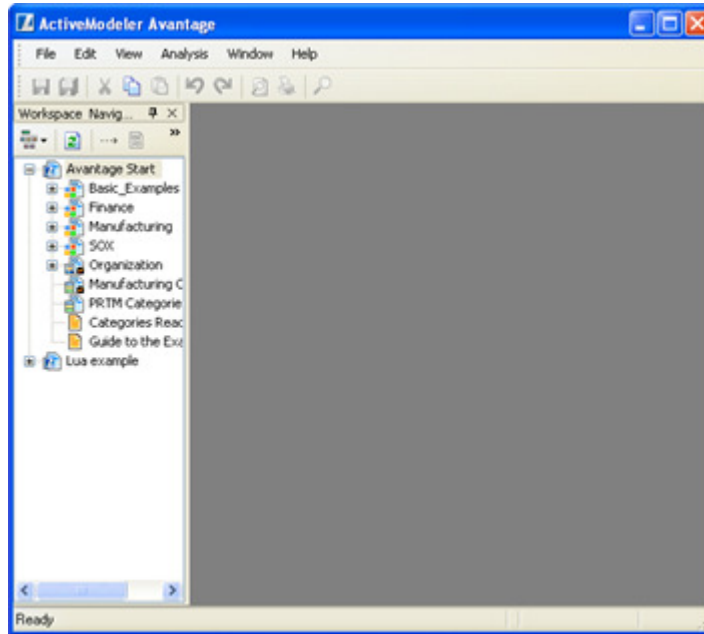


**Figure 15.** Docking to the left hand edge



Drag the mouse to the left hand docking port. As soon as it makes contact the area the window will occupy becomes highlighted in mauve. If you like this position then release the mouse and the window will dock there.

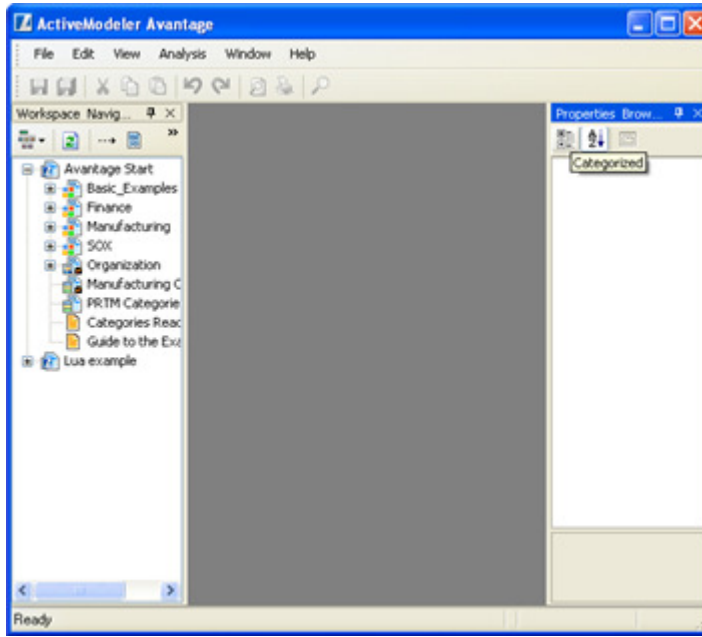
Figure 16. The Workspace Navigator after re docking



Avantage automatically remembers the layout of your views and the location on the screen where Avantage was last running. When you re-start Avantage, the Workbench state is restored to the last used state. Try it now, exit from Avantage and then immediately restart it.

Next, let's also incorporate the **Properties** browser into the picture. Select **View:Properties** from the menu bar. You should have something that looks like the Figure below.

**Figure 17.** The Properties Inspector View is now visible



**Figure 18.** Docking a window into the area already occupied by a docked window

As you move the mouse into the area occupied by a docked window you will see a new cluster of Docking ports appear.

1. Left
2. Top
3. Right
4. Bottom
5. Center (Tabbed)

Choosing the center port selects *Tabbed* docking.

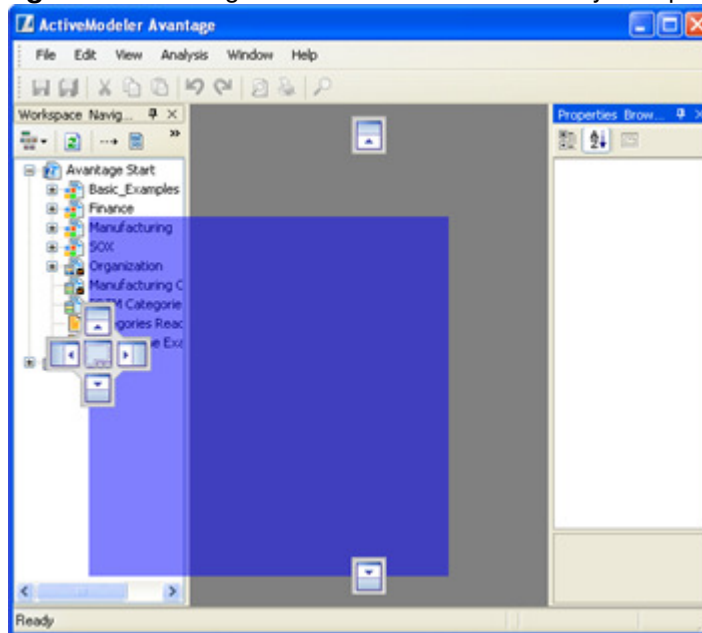
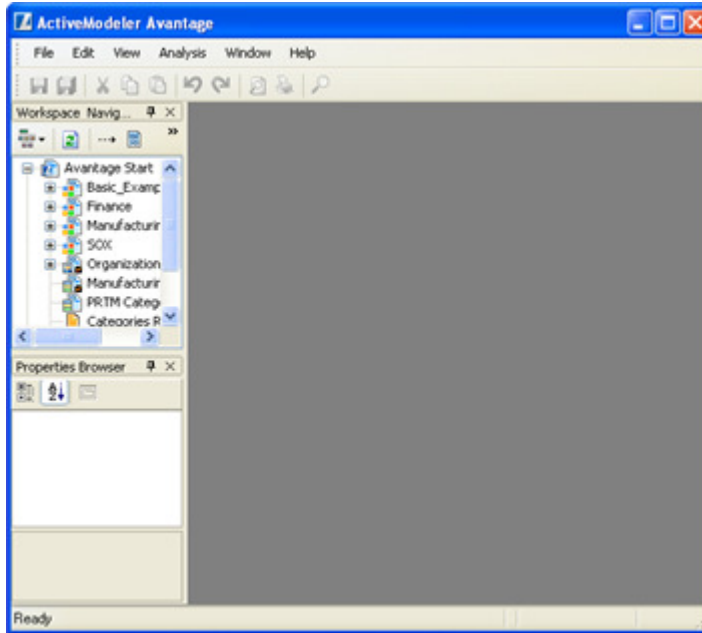




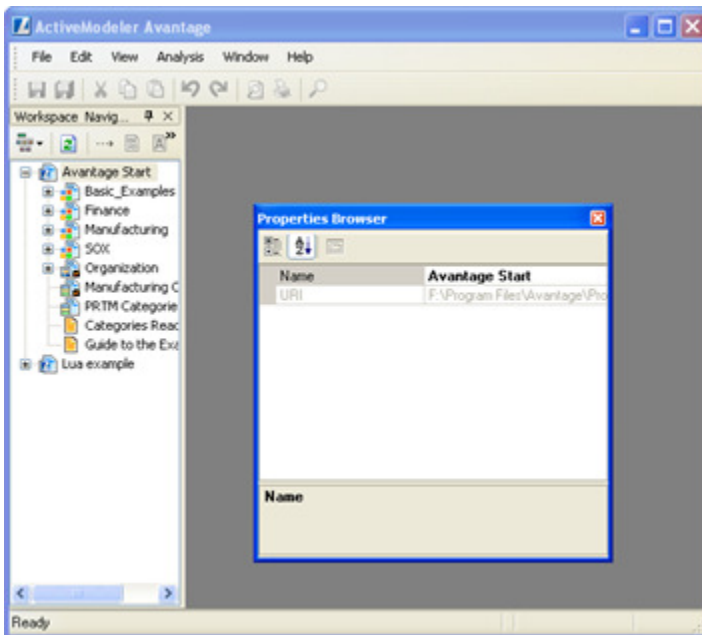
Figure 19. After docking



## Floating views

You can make a view float by dragging its title bar away from where it is docked and then releasing the mouse over the area you want it to float in.

Figure 20. Floating layout



## Tabbed views

In order to make a tabbed layout, drag a view onto another view's title bar or drag to the center docking port. Each view in the docking group is represented by a tab.

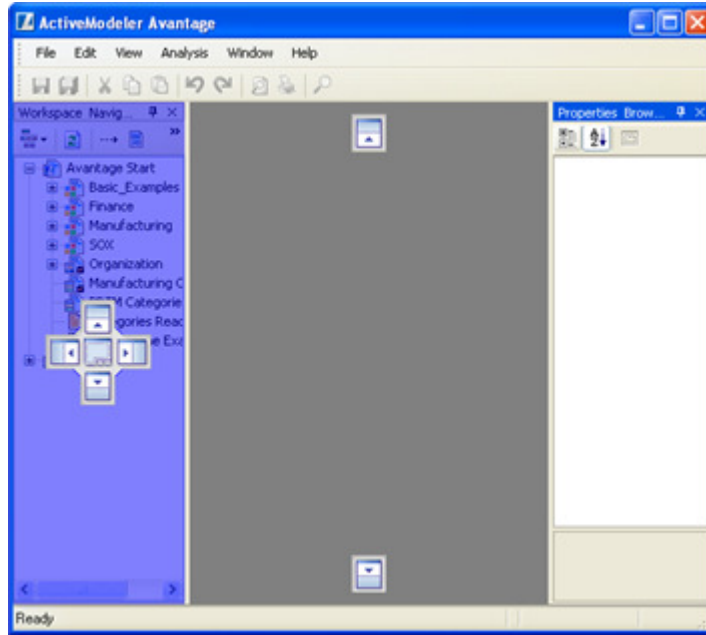
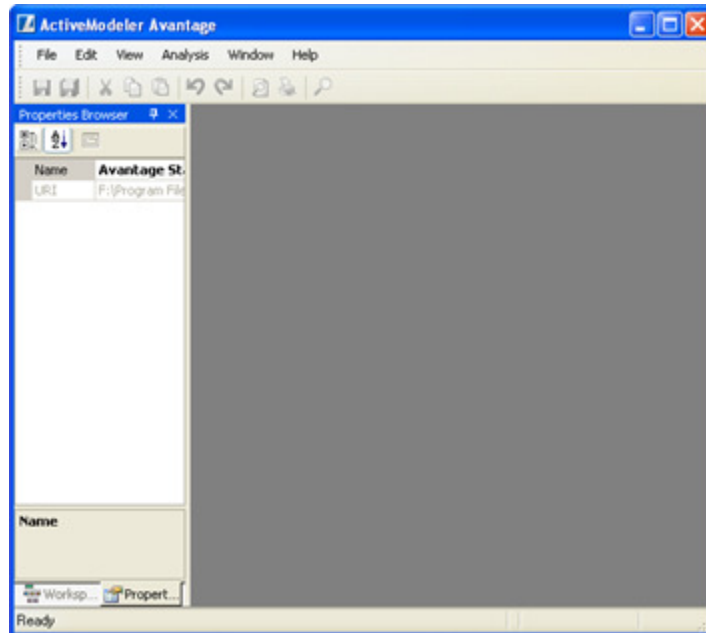


Figure 21. Tabbed layout

*When working with process diagrams it is best to ensure that both the Workspace Navigator and the Properties view are visible at the same time, so the layout in Figure 21 is not recommended.*

*(To re-dock the Properties view with the Workbench closed, drag the tab itself to the new location and release the mouse.)*



## Pinned views (auto-hide)

Dockable windows have the ability to automatically appear when needed and disappear when not. This is called “auto hide” or pinning.

**Figure 22.** A window title bar showing the auto-hide (pin) icon



Click on the pin icon to set the window into auto-hide mode. The window will immediately hide itself, showing only its title and icon. The real-estate previously occupied by the window's contents is now available for your diagram and editor windows (which will automatically expand to make use of the new space).

**Figure 23.** A pinned window

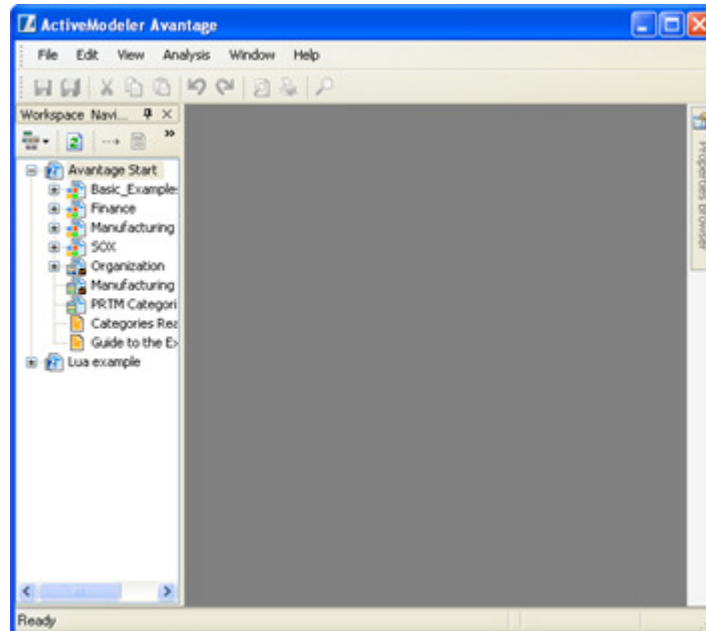
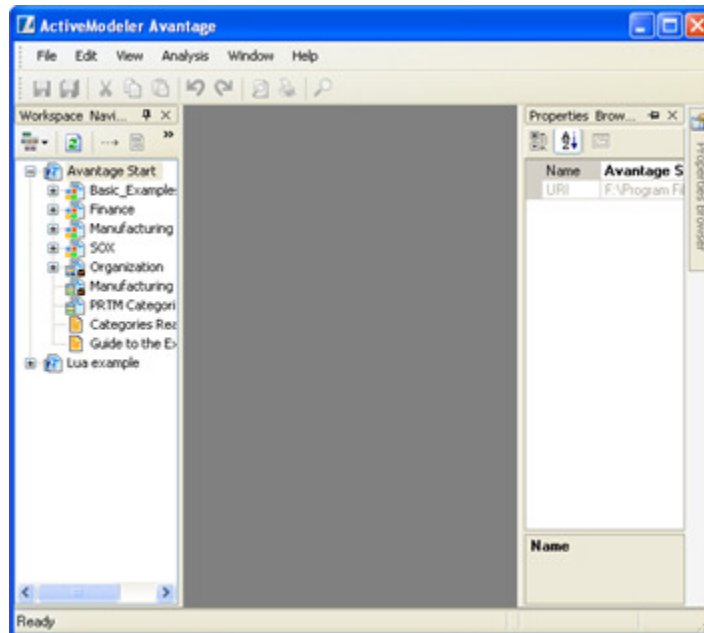


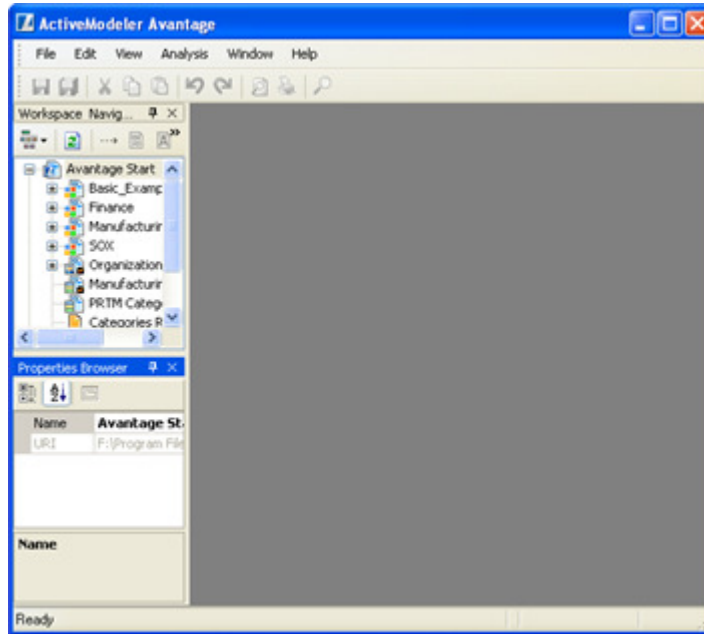
Figure 24. Showing a pinned window



To show a pinned window you just need to move the mouse over its title area. This will cause the window to automatically slide out and display itself. You can then work with it. As soon as you click in another window somewhere else it will automatically slide away and hide itself again.

## A suggested layout

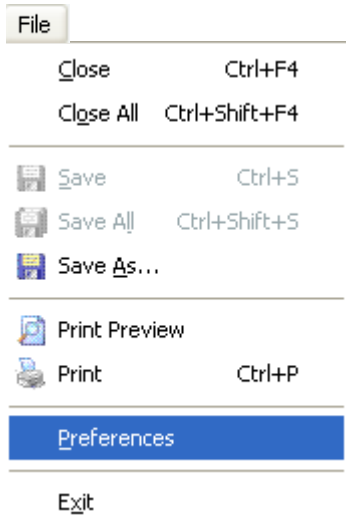
*This layout maximizes the amount of space available in the workspace to display process maps.*



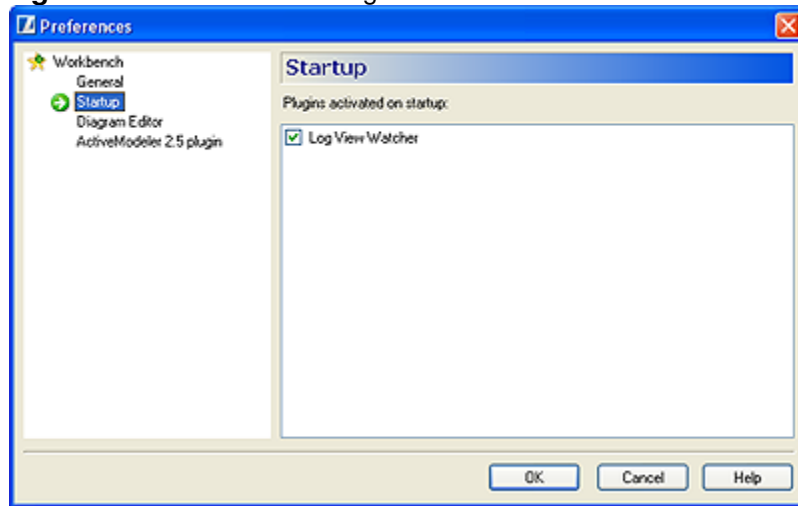
## Preferences

Avantage's own settings and those of any installed plugins can be accessed conveniently in one place through the *File-Preferences* action in the main menu bar. This action displays the *Preferences* dialog.

Click on the Workbench item to expand the list of preferences. Clicking on an item will show that item's preferences.



**Figure 25.** Preferences settings



Any registered plugins which have preferences will also appear in their own page in this dialog.

## How to Create a Project

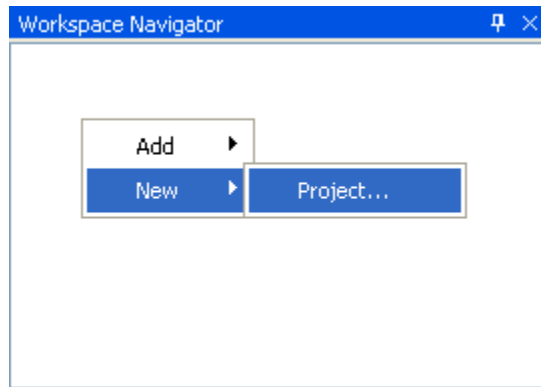
Depending on the installer you used to install the software, your initial Workspace may be empty. In which case you must create a project before you can go any further. You are probably in a hurry to try out process diagramming and ask why make a !@#Q project?

The Advantage navigator organizes your Workspace into an hierarchy of projects. You must have a project defined in order to create a process model and you must have a process model in order to create a diagram. It is usual to have multiple process models within the same project.

For more information on projects see the chapter entitled “Concepts and Terminology”.

Right hand click on the **Workspace Navigator** content area. You will see the menus that are contributed by the *Workspace Navigator* itself. Choose the *New:Project* menu command.

**Figure 26.** The New Project menu

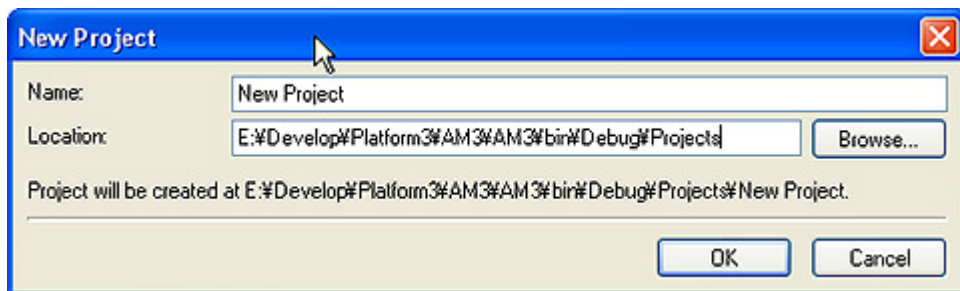


*The **Workspace** is the container of all of your Advantage **Projects**. Projects are the next level in the hierarchy from the Workspace. Projects contain files and process models.*

*Each computer user has their own Workspace*

Next, you are prompted to locate where you want the new project's folder to be created. This is the directory on your hard drive which will hold the files which comprise your project.

**Figure 27.** The new project dialog

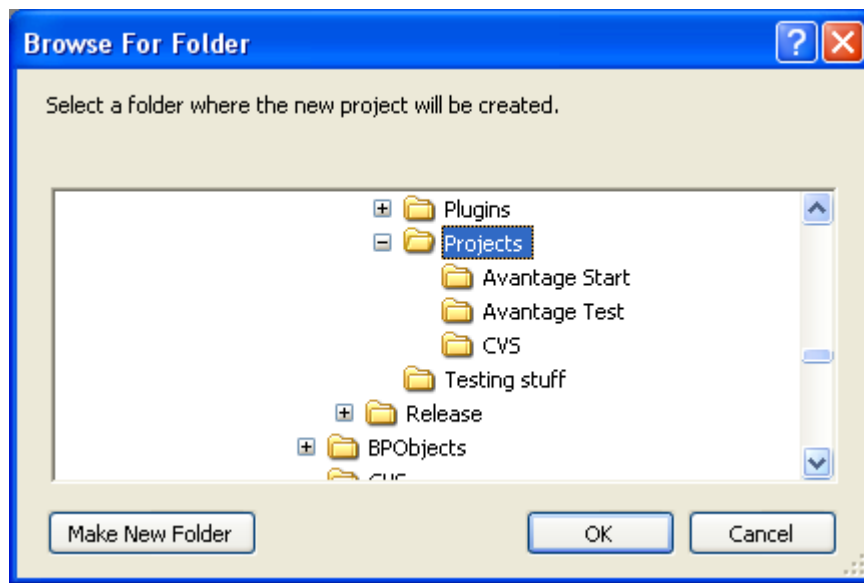


The New Project dialog allows you to specify a name and location for your new project. By default the location of a new project is the “**Projects**” folder within your workspace.

The project name becomes the name of sub-directory within the location. You can change the location by operating the **Browse...** button.

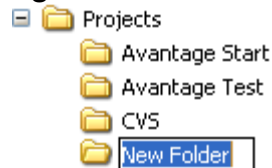
The Browse for project folder dialog starts off by displaying the Projects directory of your workspace.

**Figure 28.** The Browse for project location



Making sure that the Projects folder is selected, click the **Make New Folder** button. You will see a new folder named “New Folder” appear in the list.

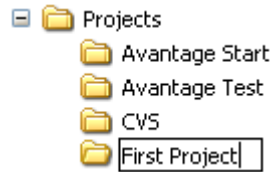
**Figure 29.** After Make New Folder



Type a name for your new project folder that is meaningful. In this example we will name the new folder “First Project”

**Figure 30.** Re-naming the new folder

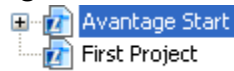




Then when you are done make sure that the new folder is selected then click the **OK** button.

At this stage Avantage will create the project and insert it into the workspace.

**Figure 31.** The new project is displayed in the workspace

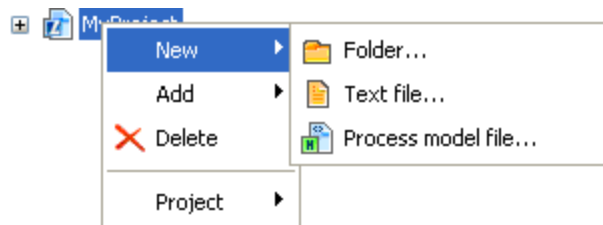


The project folder is the container where files related to the project will be created and added.

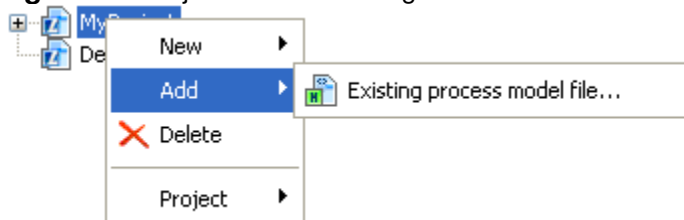
*Note:* It is possible to change the name of the project by editing the Name property of the project, however changing the name of the project will not change the name of its sub folder.

When you right-click on the newly created project you will see a context menu something like this...

**Figure 32.** Project menus showing New actions



**Figure 33.** Project menus showing Add actions



With the Project menus you can add, modify and remove *Folders*, *Text Files*, *Process model files* and other *Projects* to your project.

Because Avantage is an extensible platform you might see additional menus depending on what plugins are installed on your computer. A project is just an XML file.

**Figure 34.** Inside an *.amproject* file

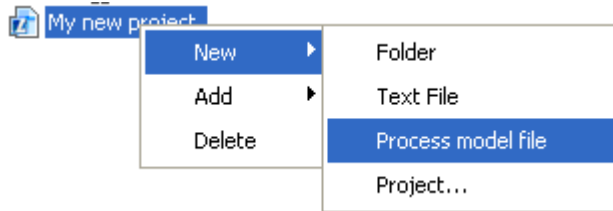
```
<?xml version="1.0"?>
<project name="MyProject">
  <files>
    <include>
      <file relPath="MyProcessModel.model" />
    </include>
  </files>
</project>
```

➔ **Note:** When Avantage opens a project it locates and verifies that all files and folders specified in the file includes list really do exist. If an entry is not found it is removed from the include list. This can happen for example if you move, rename or delete a file or folder with the Windows explorer.

## How to create a Process Model

To create a new process model file right-click on the *Project* item in the tree and select the *New:Process model file* menu action.

**Figure 35.** The new process model file menu



The New Process model action creates a *.model* file in the project's folder. The new model is named *New Process Model*. If there is already in existence a *.model* file with that name it will appear in the form *New Process Model (1)*, *New Process Model (2)* etc.

To make it easy for you to change the default name, the name of the freshly created process model file is selected into edit mode in the BPTree. You can immediately enter its proper name.

## Renaming a process model

5. Single left click on the name of the item you want to rename



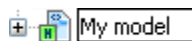
The item is now selected.

6. Single left-click on the name again so that the name is put into edit mode.

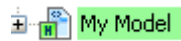


mode.

7. Enter the new name

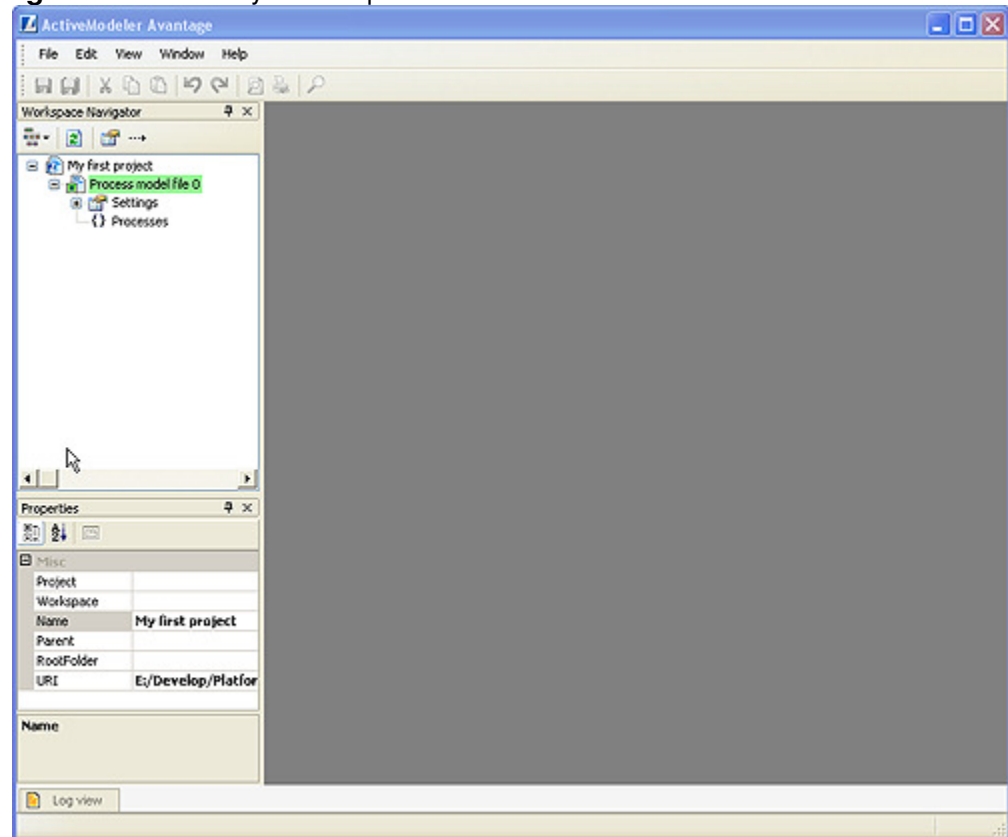


8. Hit enter to commit the change.



Expand the process model node in the Workspace Navigator to see the contents of the new process model.

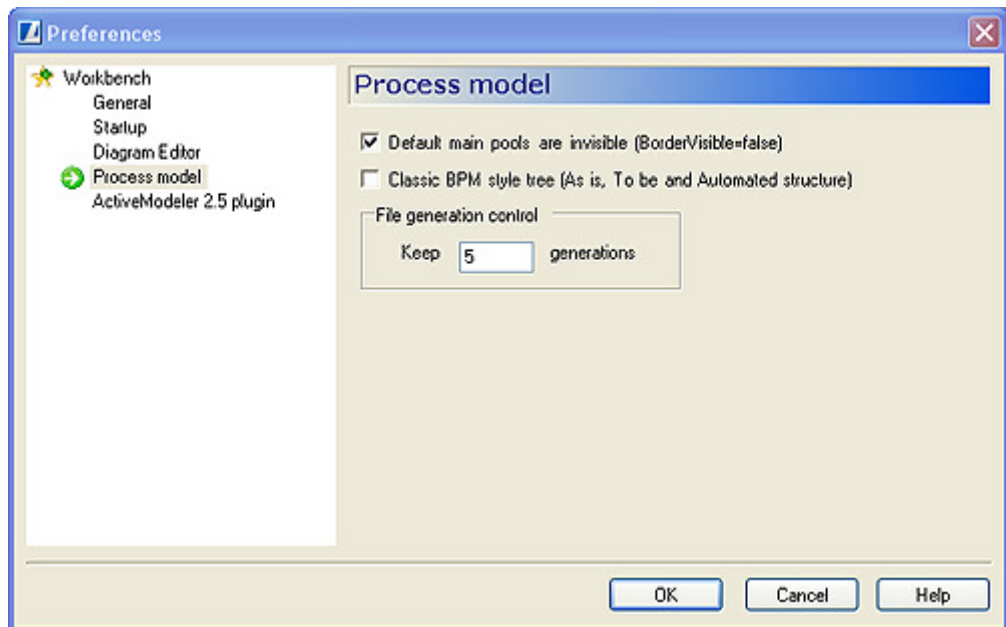
**Figure 36.** The newly created process model



## Process model options

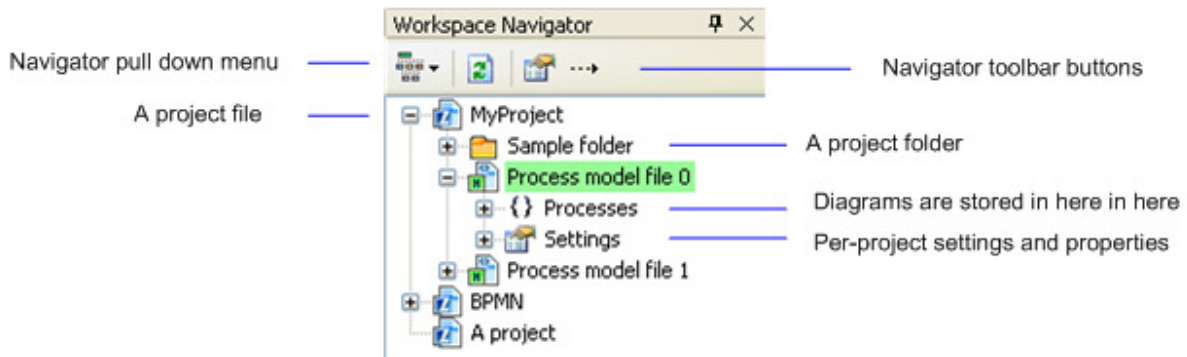
*For the simplest looking diagrams default main pools are invisible by default.*

*Classic BPM style is recommended for business analysis and process improvement modeling.*



Let's examine the process model tree in more detail.

**Figure 37.** Expanded process model



A freshly created process model contains the following components:

- ▶ Settings. This is where settings specific to this process model are displayed.
- ▶ Processes Tree  
This is where process diagrams are displayed.

## Navigator Toolbar Buttons


The navigator toolbar contains buttons for changing the amount of information that is displayed in the navigator tree.

The navigator toolbar area is a public place in Avantage and other plugins often contribute additional buttons and menus this area.



**Figure 38.** The Navigator Toolbar Buttons



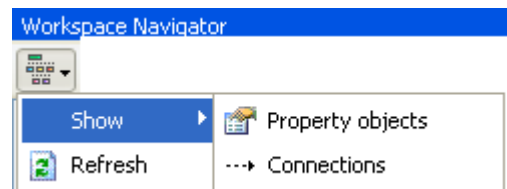
The refresh button.

 The refresh button forces the navigator to refresh the selection.

Checkbox buttons

-  Show connecting links in the tree when checked. The default is checked. Check OFF to simplify the display of the BPTree.
-  Show property objects when checked. The default is checked. Check OFF to simplify the display of the BPTree.

**Figure 39.** The Navigator View Menu



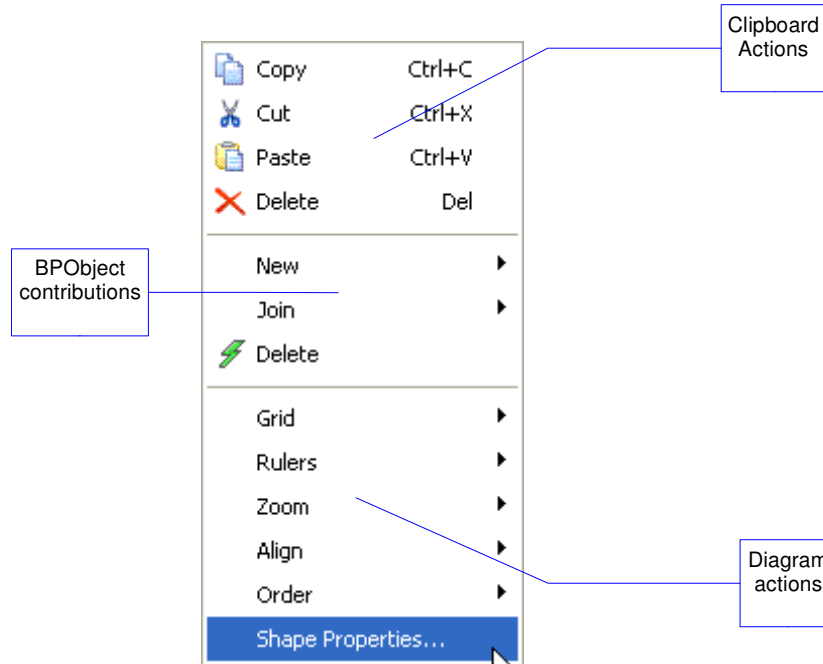
Clicking on the Navigator View menu icon displays the same buttons as in the toolbar buttons. It is just an alternative method. The Refresh button can be used to refresh the Navigator view.

## Object actions

All BPObjets possess a set of menu accessible actions known as *contributions*. Contributions are visible to you when you right-click on a BPObjct wherever it is. It does not matter if you right-click on a shape representing a BPObjct within a diagram editor window or right-click directly on an object itself in the Navigator, the same *contributions* are displayed to you. It is convenient to think of these built-in actions as process model actions.

**Figure 40.** The various contributions to the context menu

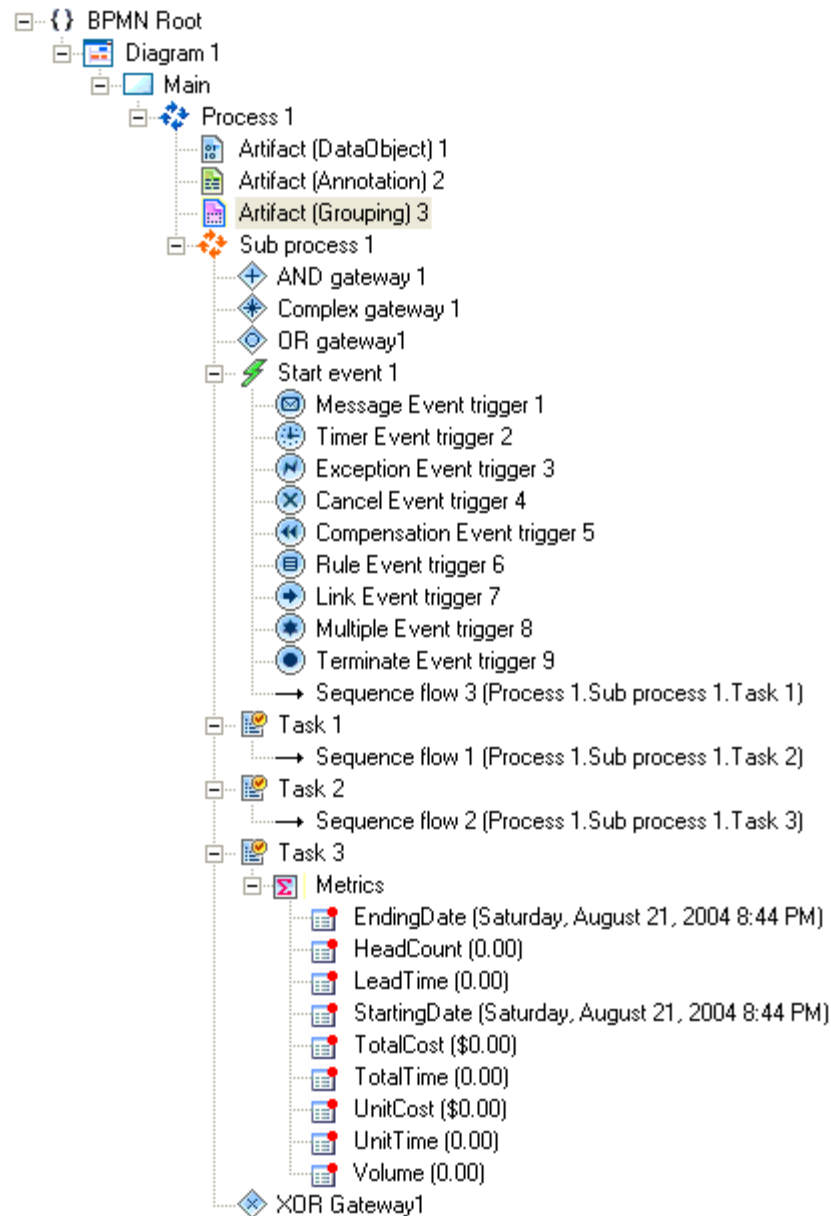
*This screen shot shows the division of labor within diagram editor menus. The top group of menus manage the clipboard. The middle group of menus manages the object contributions (which are common to both the BPTree and the diagram editor) and the bottom group manage the diagram editor menus which apply only to the diagram editor.*



## Navigator Symbols

The complete repertoire of BPMN objects that can be displayed in the Workspace Navigator are shown in the figure below. The BPMN 1.0 specification is implemented in Avantage.

**Figure 41.** Symbols repertoire





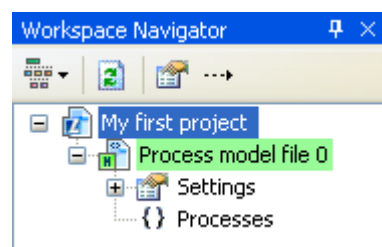
Metrics are a KAISHA-Tec extension which provide base support for capturing input for the **Metrics and Analysis Plugin** (an additional extra not included in the basic Avantage Community Edition modeler).

## To create a process diagram

To commence editing a process diagram you must first locate the line in the BPTree which contains the Diagram object you want to edit and then double-click on it.

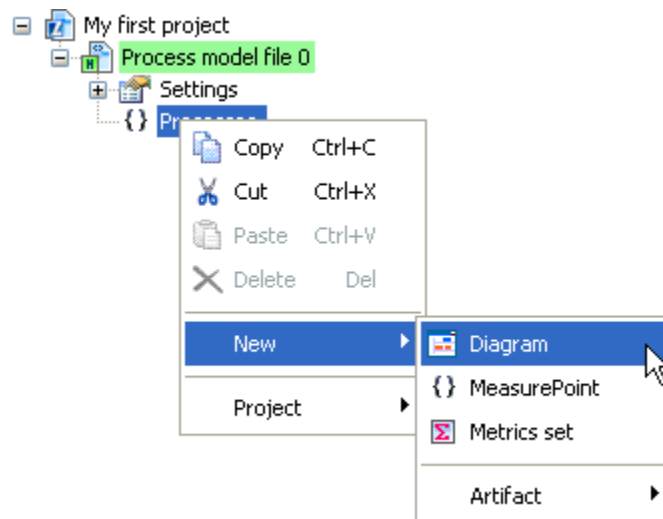
In our case, we have just created a brand new process model, so it has no diagrams defined. We have to create a process diagram object before we can edit one with the diagram editor.

**Figure 42.** Empty BPTree



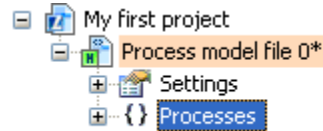
Right-click on the Processes line in the BPTree you will see a menu appear.

**Figure 43.** Creating a diagram



Choose *New:Diagram* from the context menu. After you release the mouse you should see something like below:

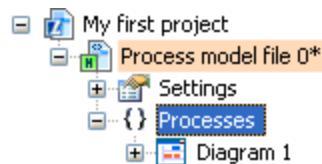
**Figure 44.** After the New Diagram command.



Notice that now there is a + (plus) sign in front of the *Processes* line in the BPTree. Also, notice that there is now a \* (asterisk) character drawn after the project's name and the highlight color is no longer green. This indicates that the project has been changed.

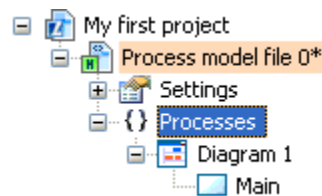
This + (plus) sign indicates that there is now something *contained* within the processes. Click on the plus sign in order to expand the tree. You should see something like this....

**Figure 45.** After expanding the Processes line



Now we can see that there is a Diagram with a title of “Diagram 1” in the BPTree. If you look carefully you can see that it too has a + (plus) sign in front of it. Click on this plus sign to see what is inside...

**Figure 46.** After expanding the diagram line



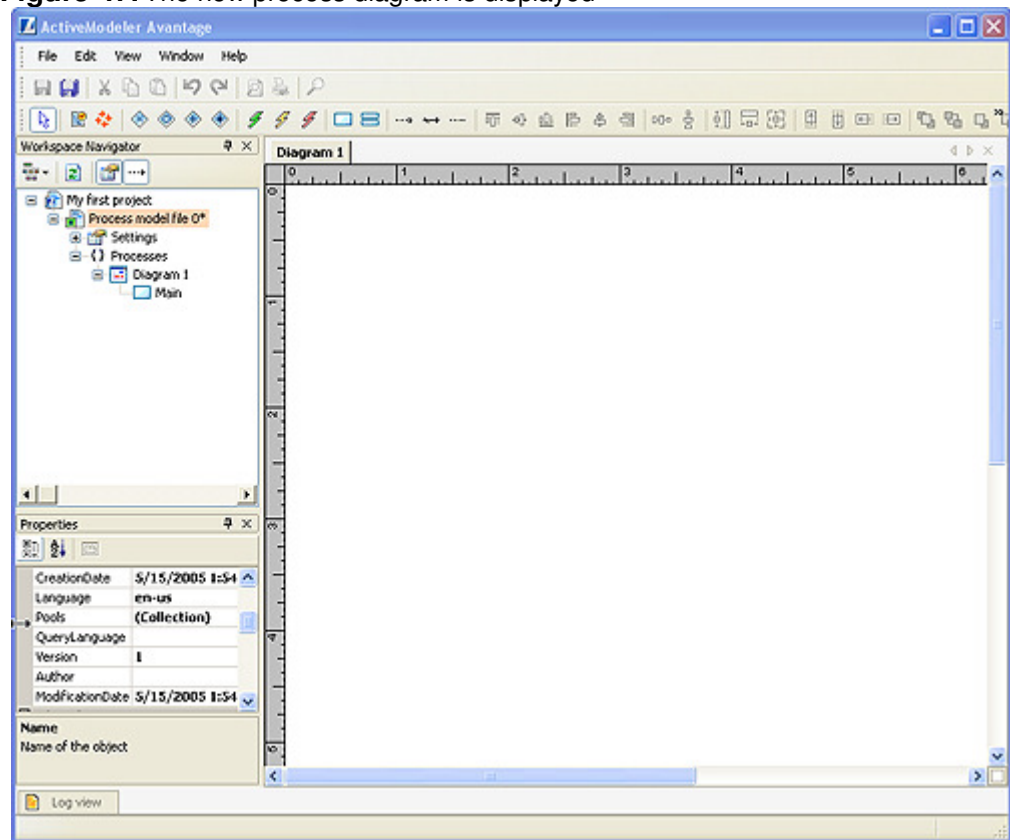
Now we can see that there is a funny looking shaded blue rectangle called “Main” inside the diagram object. This object is a *Pool* (something for *Tasks* to “swim” in).

One of these things (Pools) is created automatically whenever you create a new diagram.

OK. Now double-click on either the Pool or the diagram line in the BPTree.

When you double-click on a flow object in the BPTree Avantage will open its enclosing diagram object in a diagram editor window. You should see a diagram window called “Diagram 1”.

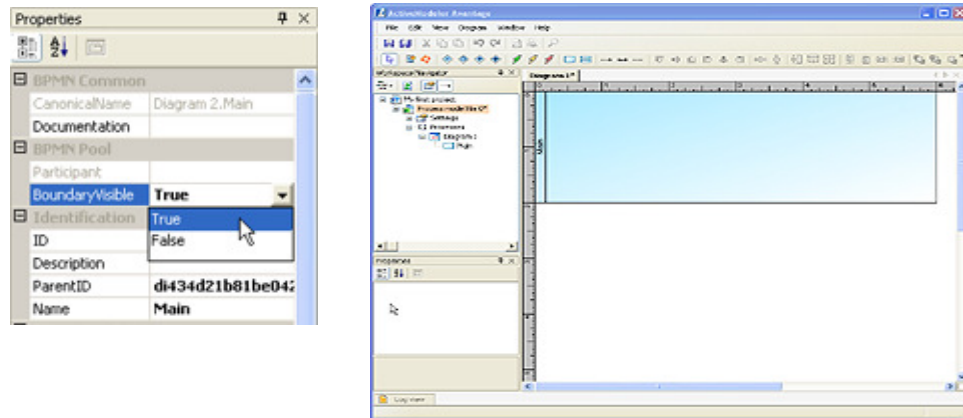
**Figure 47.** The new process diagram is displayed



Notice something strange? The *Pool* object called “Main” which is plainly visible in the BPTree is not shown in the diagram editor. Why is this? The BPMN standard says that diagrams have a default border less pool into which process flow objects are put by default Avantage doesn't show this pool..

If you need to see the default pool then you can easily do so. Single click on the “Main” entry in the BPTree. Notice that the *Properties* browser immediately shows the properties of this pool.

**Figure 48.** Set *BoundaryVisible* **True** to make the pool visible in the canvas



To make the pool visible click the *BoundaryVisible* drop-down and select True.

## Placing flow objects in a diagram

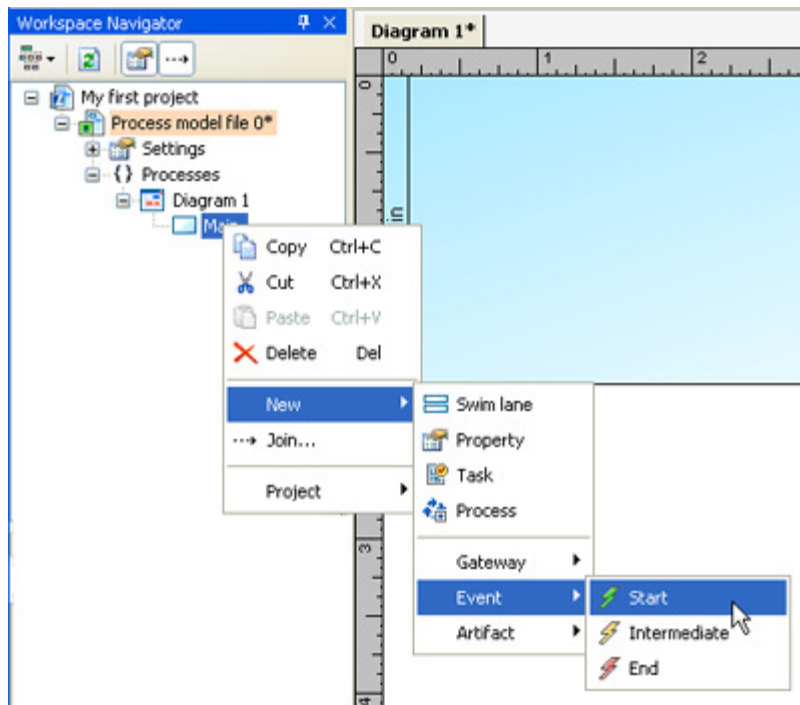
We will now by way of example create an *Event* object and place it in the diagram. An Event is something that “happens” during the course of a business process. These Events affect the flow of the Process and usually have a cause or an impact. Start and most Intermediate Events have “Triggers” that define the cause for the event. There are multiple ways that these events can be triggered.

Enough theory, we will create a StartEvent which will signify the beginning of an activity in our workflow. The trigger for starting will be a Message.

Note: Although it is usual to edit BPMN diagrams using the BPMN toolbar it is also possible edit BPMN objects directly by using the context menus which are available from an object (available process diagrams and the navigator view).

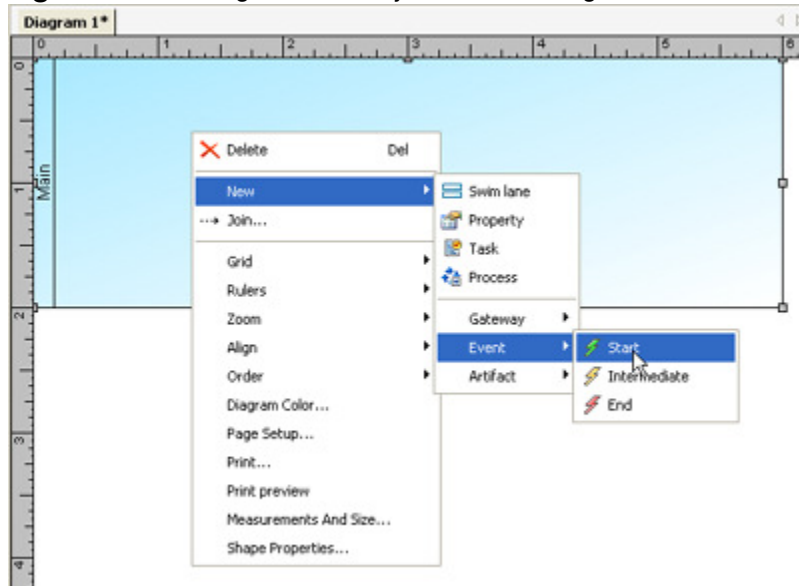
The first example (creating a StartEvent) will demonstrate how to use context menus from the navigator, the diagram editor as well as the BPMN toolbar.

**Figure 49.** Creating an event object with the BPTree



- *Right-click on the Main pool icon in the BPTree.*
- *Choose New*
- *Choose Event*
- *Choose Start*

**Figure 50.** Creating an event object with the Diagram editor



- *Right-click on the surface of the Pool shape in the diagram window*
- *Choose New*
- *Choose Event*
- *Choose Start*

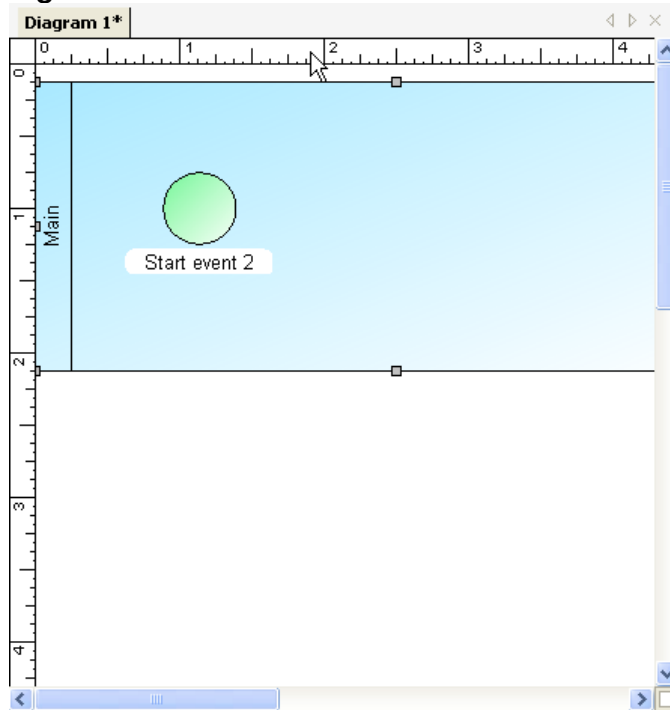
Both methods achieve the same result. A *StartEvent* is created and placed in the Pool.

**Figure 51.** Using the BPMN toolbar



The BPMN toolbar is the easiest way to create the event shape. Firstly, ensure that the Pool is the currently selected object in the diagram (this ensures that it becomes the parent/container). Click on the green StartEvent button (highlighted in this figure) to arm the toolbar and then click in diagram window at the position where you want the new event shape to be displayed.

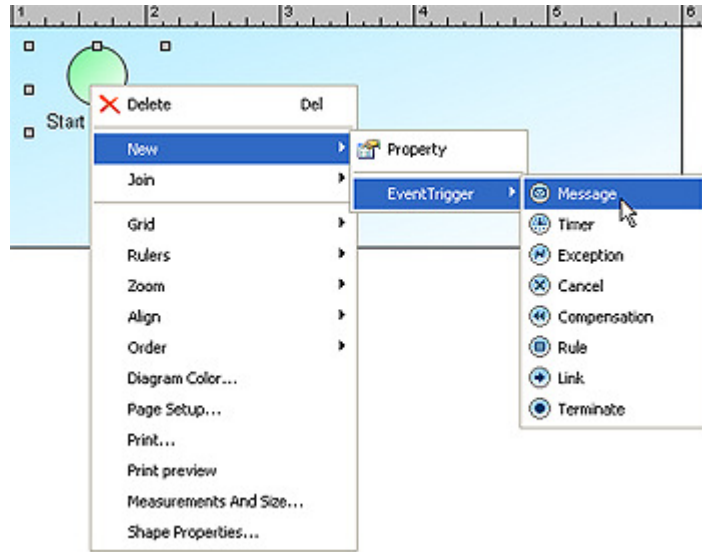
**Figure 52.** Result of the create event action.



*Note:* If however, we didn't choose to make the “Main” pool visible it would not have been possible to insert the event from the diagram editor since there would have been no shape in the canvas to click on.

Event objects have *Triggers*. A trigger describes what causes the event to be activated. The newly created event in the screen shot above requires an *EventTrigger* before it is complete.

**Figure 53.** Adding a trigger from the diagram editor



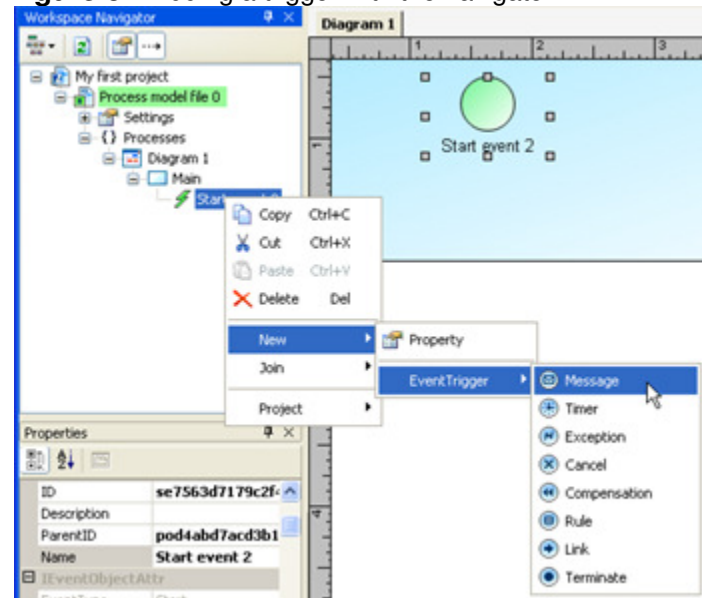
•Right-click on the StartEvent

•Choose New

•ChooseEventTrigger

•Choose Message

**Figure 54.** Adding a trigger with the navigator



•Right-click on the Event object

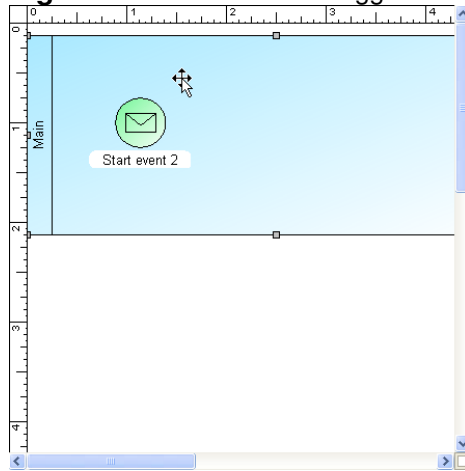
•Choose New

•ChooseEventTrigger

•Choose Message



**Figure 55.** The event with trigger is displayed



You can drag the shape around the canvas, change its color, change its line properties, font properties and more.

## Changing a Shape's properties

**Figure 56.** The Shape Properties command

For any shape in the diagram editor you can change its graphical properties (like Font, Fill color, line width etc).

Right-click on the *Shape Properties...* menu.

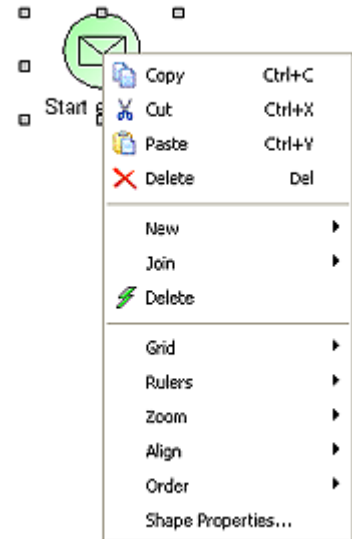
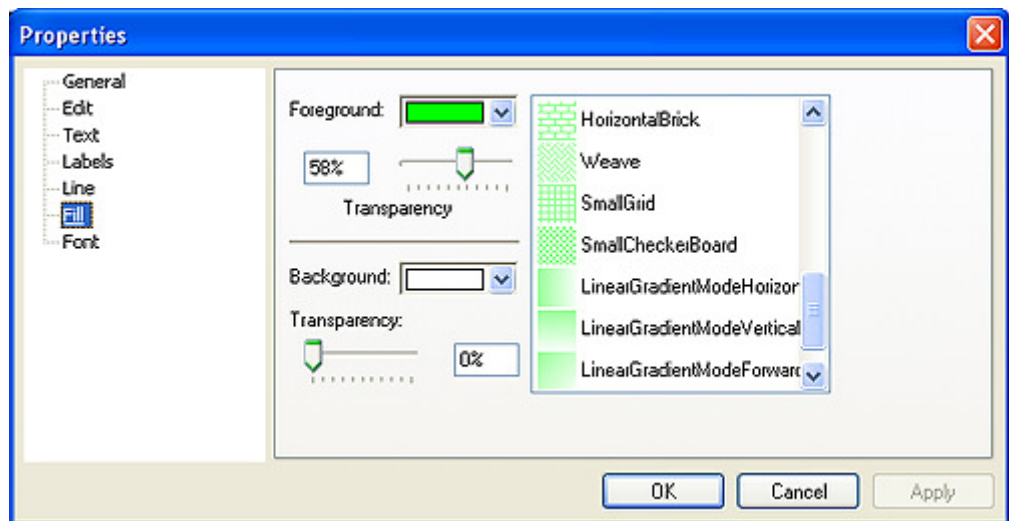
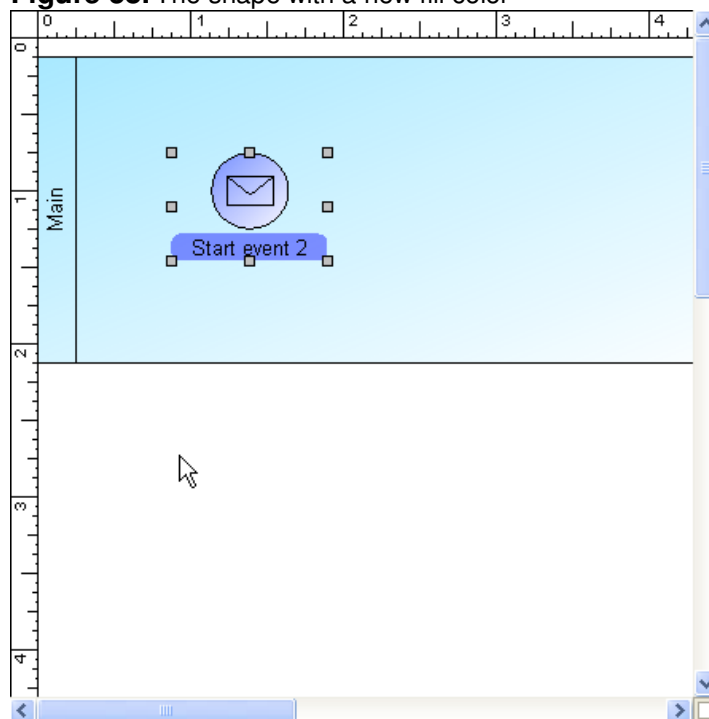


Figure 57. Shape properties dialog



Click on the *Fill* property, this will show the various fill settings for the shape. In this example we will change the shape's color. Click on the *Foreground* drop down and select some other color, then finally click **OK** to dismiss the properties dialog.

Figure 58. The shape with a new fill color

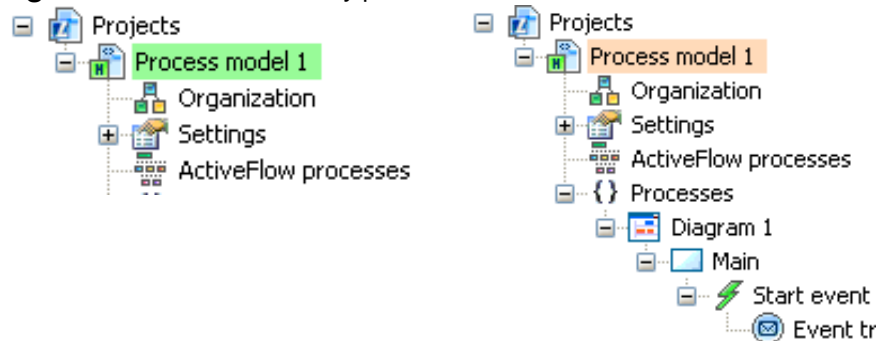


## Saving and coordinating changes

After you have made changes to the process model, either with the diagram editor or the BPTree the process model will be flagged as “dirty”.

You can easily tell that the model is dirty. The name of the model has an \* (asterisk character) appended to it. As well, the background colour of the name is drawn with a different colour..

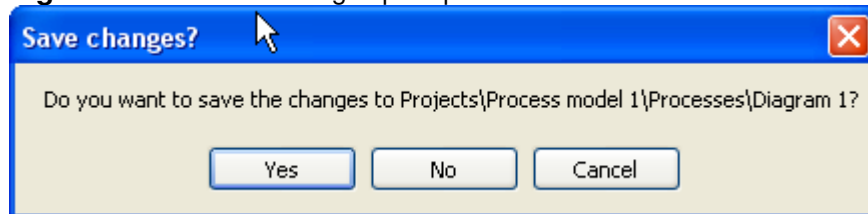
**Figure 59.** Clean versus dirty process model file



A clean (unmodified) process model is highlighted with a green background, a dirty process model is highlighted with a peach colored background.

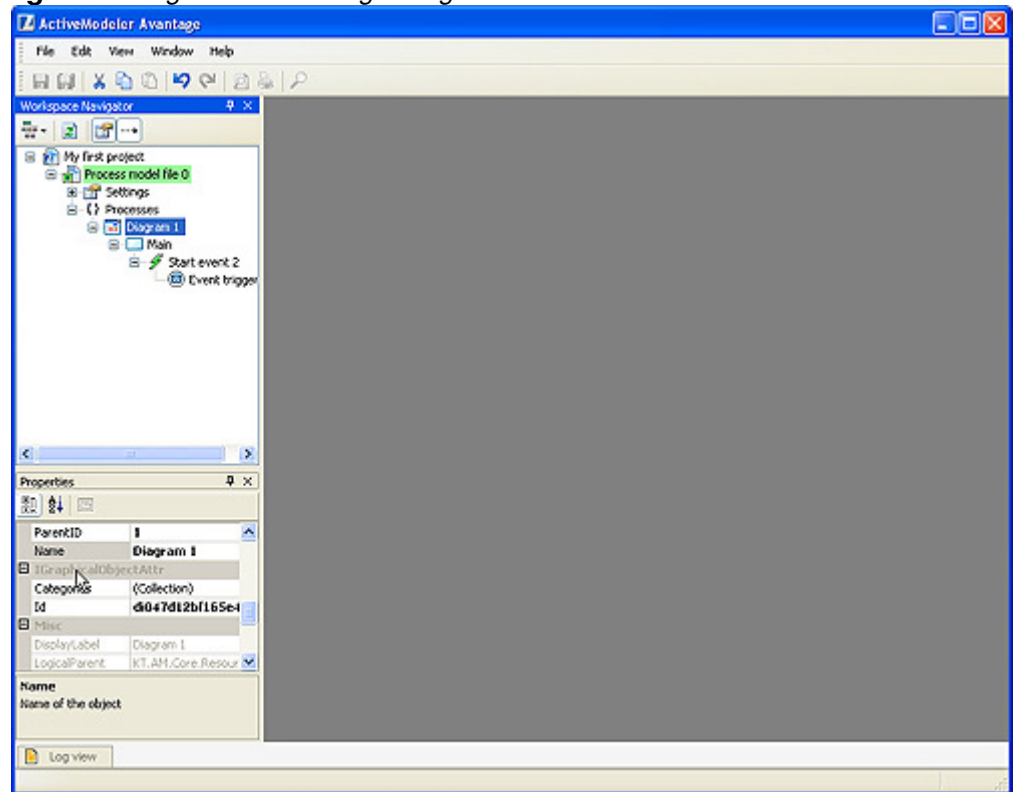
If you close the diagram editor window Avantage will ask you if you want to save any changes that you have made.

**Figure 60.** The Save changes prompter



Click on **Yes** in order to save your changes.

**Figure 61.** *Figure: After saving changes*



After the process model has been saved its title is underlined in green (to indicate that it is clean).

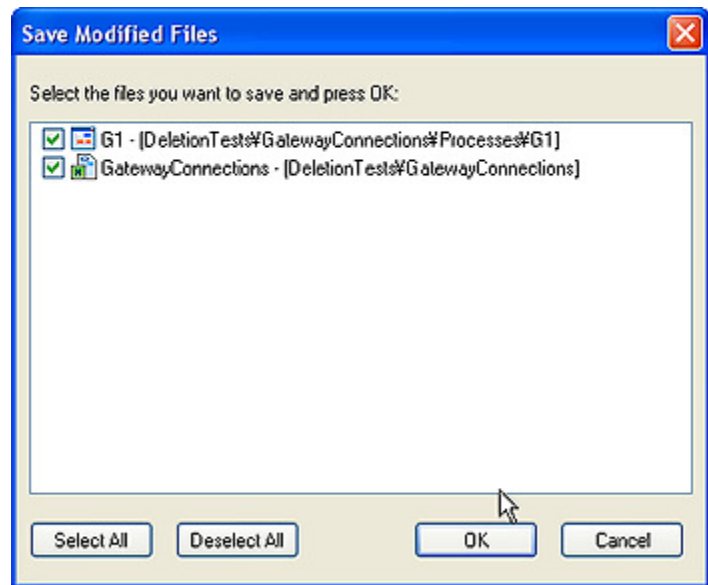
## Saving files at exit

When you exit Avantage you are prompted to save any changed open files with this dialog.

By default all the items are checked ON. If you don't want to save a particular item check it OFF.

## The Save action

The *Save* action is different depending on which plugin's window is active when the action is executed.



### For Diagram windows

Save (**Ctrl + S**) works by copying the canvas level (as well as any structural) changes back to the process model file.

### For the Workspace Navigator window

The Save (**Ctrl + S**) action updates the process model file which is currently in the Workspace Navigator selection. Any project model related selection is sufficient to allow the Save action to be active. (Any selection of type `.model` or of type `BPObject` belonging to a `.model`).

The Save action is enabled if the selected process model file is dirty. Saving resets the file back to clean (the highlight color is green).

### Forcibly saving a process model



You can forcibly save a process model file by right-clicking on its Settings icon and choosing the **Forced save** action. It does not matter if the file is dirty or not – the file will be forcibly saved.

## The Save As action

### For Diagram windows

Avantage process diagrams are not stored in individual files. Process diagrams are just a specific kind of `BPObject` which lives in the process model repository along with all the other `BPObjects` in the repository.

Consequently the Save as action saves the diagram as a `.section` file (which is a `BPObject` exchange format). A `.section` can be imported into a process model.

## For Workspace Navigator

The Save As action allows you to save the current .model (repository file) to another file destination. This action does not effect the repository file of the active process model in any way, the current file is merely copied to the Saved as path.

You can also Save the current selection as Section (a *.section* file).

## Process model Revert

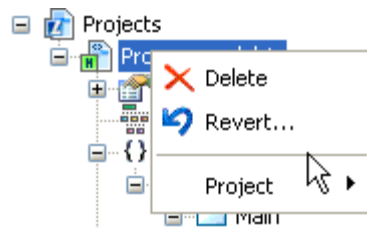
In the event that you make a mistake when editing a process model and you don't realize the error until after the process model file has been saved it is not a problem. This is because it is easy to **Revert** back to a previous good version and resume your work.

Avantage uses a rotating backup system. By default, Avantage keeps 5 previous generations which are named in the manner of \*.model, \*.model.1, \*.model.2, \*.model.3, up to \*.model.5. This allows the process model to be reverted to one of 5 most recently saved copies in the event of a file corruption, IO error or other accident.

## How to Revert to a previous generation

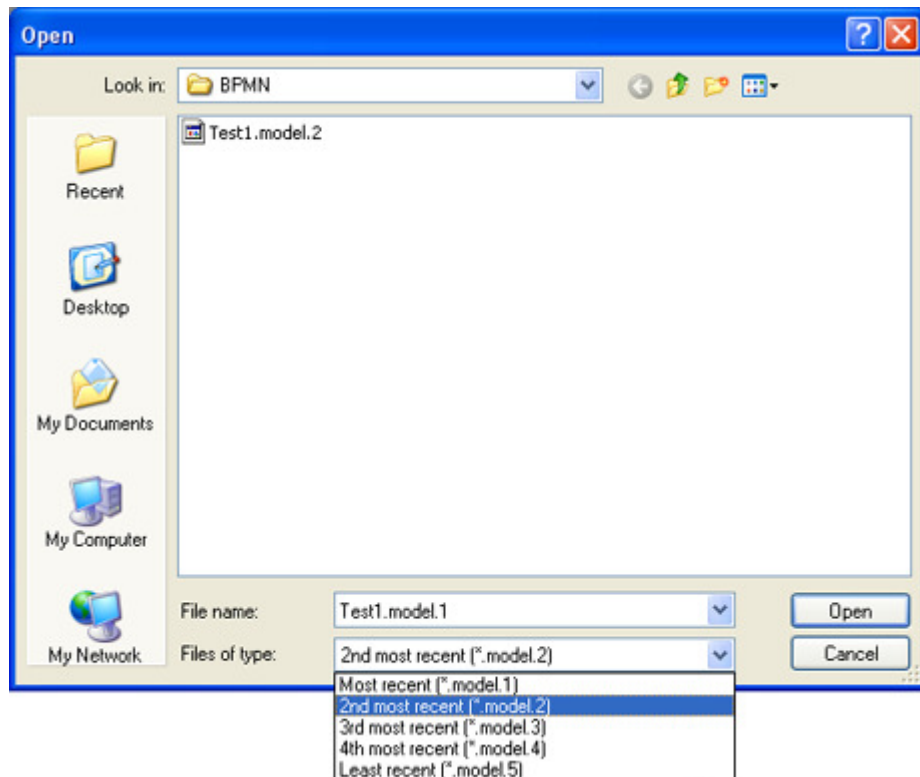
Right-click on the process model file you want to restore. Choose the Revert... action like in the figure below.

**Figure 62.** The revert action



The file chooser dialog allows you to select between generations.

**Figure 63.** Choosing a generation file to revert from



After you select the generation file and click OK, Avantage checks the file to determine if it is suitable. Take heed of warning messages since it is possible to restore an unrelated process model file if you ignore the warnings.

## The different types of modifications/changes

### Structural changes

Changes made with object the BPMN Toolbar, object contribution menus operate directly on the process model in-memory. The process model may be connected to *listeners* (for example the Properties browser and Diagram editor windows are *listeners*).

When a notifiable event (such as a modification an addition or a deletion) occurs in the process model then all registered *listeners* are automatically notified. Your diagram window will be automatically updated as a result.

The diagram editor can make structural changes to the process model. The BPMN Toolbar can also make structural changes.

### Drawing canvas changes

Changes made with diagram editor such as dragging a shape to a new location on the canvas, changing the color of a shape and others operate on the presentation layer of data which is managed by the diagram editor. The diagram editor accumulates all of these changes locally until such time as the diagram is saved.

It is this second group of changes that the diagram editor is prompting us to save when we closed the diagram window. When we answer Yes to the prompt, the diagram window copies those changes to the process model and then rewrites the *.model* file.

When a diagram window is open and you can also make structural changes to the process model with the object contribution menus of the BPTree and these changes are immediately visible in the diagram window. For example, let's say we are editing a diagram and from the BPTree we insert a new *Task* into the main pool. This task object will be shown in the diagram window immediately.

Similarly, if you create an object with the BPMN toolbar, or use an object contribution menu within the diagram editor the process tree in the Navigator is updated to show the new object.



## Clipboard operations

Both the BPTree and the diagram window support the common clipboard operations.

### Navigator (BPTree) clipboard operations

The BPTree copies data to the clipboard in a format known as a *BP\_Section*. A section is a self contained slice of a process model. A section has a *root object* which is the active selection in the Workspace Navigator/BPTree at the time the copy or cut action was executed (this is *serialized* as the first <BPTreeObjects> entry in the BP\_Section Xml as demonstrated in the figure below).

In the table below that data was created by selecting a single Task object in the Navigator BPTree and choosing the **Copy (Ctl+C)** menu command.

**Figure 64.** A BP\_Section with a single item

```

<?xml version="1.0" encoding="utf-8"?>
<BPObject xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="BPTreeSerializer">
  <Expanded>false</Expanded>
  <ID>85d6c6a5024f453ca57cf2ff891e72e6</ID>
  <Persistence />
  <Name />
  <IsStructureLocked>false</IsStructureLocked>
  <IsValueLocked>false</IsValueLocked>
  <IsAttributeLocked>false</IsAttributeLocked>
  <BPTreeObjects>
    <anyType xsi:type="BPD_Task">
      <Expanded>false</Expanded>
      <ID>tae314d64b3a4a448eb7bcc4c86494a7b8</ID>
      <Persistence />
      <Name>Task 1</Name>
      <ParentID>po5d70a58b18624a77b150f5cc5cde309d</ParentID>
      <IsStructureLocked>false</IsStructureLocked>
      <IsValueLocked>false</IsValueLocked>
      <IsAttributeLocked>false</IsAttributeLocked>
      <PoolID>po5d70a58b18624a77b150f5cc5cde309d</PoolID>
      <LoopCounter>0</LoopCounter>
      <LoopMaximum>0</LoopMaximum>
      <MIOrdering>Sequential</MIOrdering>
      <MIFlowCondition>All</MIFlowCondition>
      <TestTime>Before</TestTime>
      <AssignTime>Start</AssignTime>
      <TaskType>None</TaskType>
      <Status>None</Status>
      <StartQuantity>1</StartQuantity>
      <LoopType>None</LoopType>
    </anyType>
  </BPTreeObjects>
</BPObject>

```

Graphical per-shape properties related to the diagram editor (such font, fill color, line properties etc) are stored in the BPObject.Persistence property. In this example section Persistence is blank because the task was just freshly created from the BPTree and had not yet been edited with the diagram editor.

## Paste and paste targets

When you paste a section, the Navigator attempts to insert the section as a child of the *current selection*.

Earlier we explained how each *BP\_Section* has a *root* object. This root object is of a certain type (in our example above the root object type is *BPD\_Task* )

In the BPMN standardized order of things a *Task* can only be contained within certain kinds of flow objects. It is illegal for a Task contain another Task as its child. If for example you tried this misoperation Avantage would display an error message about the wrong kind of parent.

## Copying objects from one process model to another.

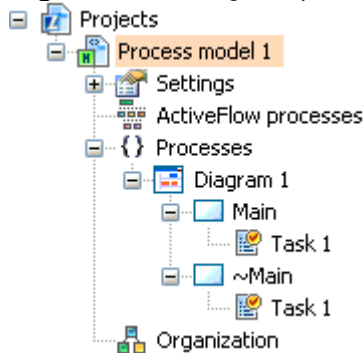
You can use the Copy/Paste sequence to move data between process models, even between process models in different projects.

## Copying objects from one part of a process model to another

You can use the Copy/Paste sequence to effect a move operation. As long as you paste the section into an appropriate container object then the operation will succeed.

BPObjects (hierarchical process model objects) are identified uniquely in a process model by their **ID** property. If the **ID** of an object contained in a section clashes with that of an object of the same type with the same **ID** in the process model that is receiving the section then the ID will be changed. In order to provide you with a visual cue that something is different the pasted object's name is prefixed with the “~” character. This effectively means “Copy of “ but is a briefer and easier to decipher visually.

**Figure 65.** Pasting a duplicate object



*Here a Pool object was selected and copied. Then the diagram object was selected and the Paste action was executed.*

*The ID of Main was already registered within the Diagram object's children so it was given a new ID property and the name ~Main shows that it was a duplicate when it was pasted.*

Regardless of ID clash resolution, if an object of the same type as that of the pasted object having the same name already exists as a sibling, then its name is also prefixed by a “~”.

## Process flow connecting objects

There are three kinds of connecting links that can appear in a process diagram, these are: *Sequence flows*, *Message flows* and *Associations*.

### Sequence flows

A *Sequence Flow* is used to show the order that activities will be performed in a Process. Each flow has only one source and only one target. The source and target must be from the set of the following flow objects: *Events* (*Start*, *Intermediate*, and *End*), *Activities* (*Task* and *Sub-Process*), and *Gateways*.

### Message flows

A *Message Flow* is used to show the flow of messages between two entities that are prepared to send and receive them.

### Associations

An *Association* is used to associate information and *Artifacts* with flow objects. Text and graphical non-Flow Objects can be associated with the flow objects. An Association is also used to show the activities used to compensate for an activity.

## How to create connection objects

You can create connections with the BPMN toolbar or through the context menus of the Navigator and the diagram editor.

### Creating connections with the BPMN toolbar

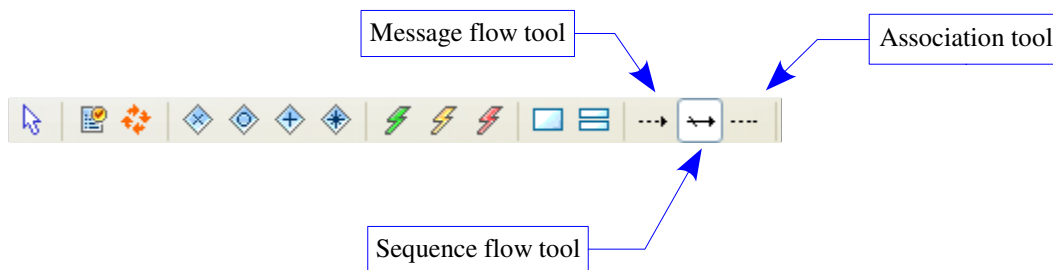
Clicking on a linking tool arms the diagram editor to link between two flow objects. The mouse cursor changes from a pointer to a small cross-hair to indicate that the action is armed. The mouse cursor changes to a cross-hair enclosed within a circle when the object the mouse is over can be a link source or link target.

➔ **Note:** Because BPMN allows you to create connections to Swimlanes and Pools you must be careful to move the mouse cursor inside of the shape of the target object you intend to link with.

For detailed instructions see the Diagram Editor chapter.

Release the mouse to complete the connection.

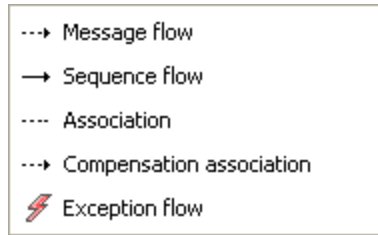
**Figure 66.** BPMN Toolbar connection tools



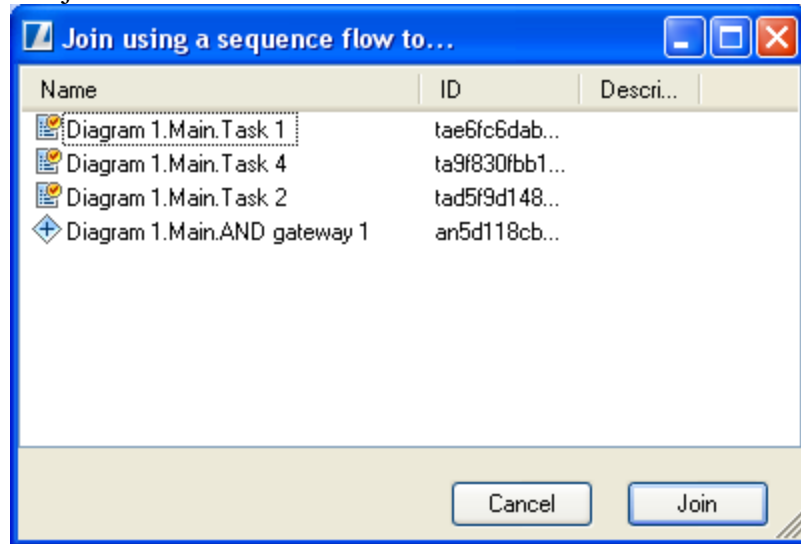
### Creating connections using context menus.

The principal is the same, irrespective of the type of flow object you are working with. You can join flow objects with either the Navigator/BPTree or the diagram editor window by using the object contribution menus which popup when you right-click on them.

1. Right-click on the *source* flow object.
2. Select the *Join* menu from the popup
3. Choose the connection type that you want to create (Sequence, Message, Association, Compensation or Exception flows).



4. A dialog box will appear that allows you to choose a suitable *Target* object.



5. Select the target object from the candidate list in the dialog and then click the “*Join*” button.

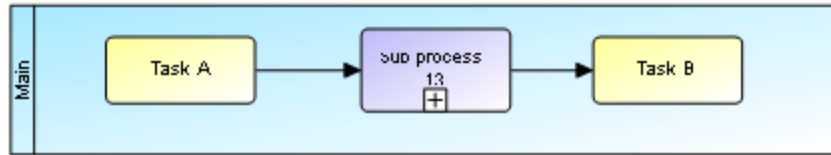
For more detailed instructions on creating joins using the context menus please see the Navigator Actions chapter of this manual.

## Nested Process Descriptions (Sub maps)

Most process modeling tools provide the ability to link an activity or task to another process. You will variously hear the use of terms like “subprocess”, “sub map” etc. These are examples of nested process descriptions.

BPMN provides this functionality through the SubProcess object.

**Figure 67.** Process flow containing a SubProcess



A SubProcess may contain either:

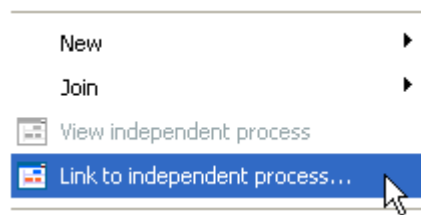
1. An external/independent process flow. In this type of sub process, the SubProcess object contains a link to a separate Diagram. (*Note:* The diagram must be in the same process model)
2. An embedded process flow. In this type of sub process, the objects representing the embedded process are stored within the host SubProcess object..

To display either kind of sub process click the “+” indicator of the SubProcess object. If the subprocess links to an independent diagram then a new diagram window is opened. If the SubProcess is the embedded type then the SubProcess shape will automatically expand out in size to reveal its contents. These are known as *Expandos*.

## Independent (external) sub processes

To create a link to another process diagram select the SubProcess object then right click on it to choose the “Link to independent process...” action.

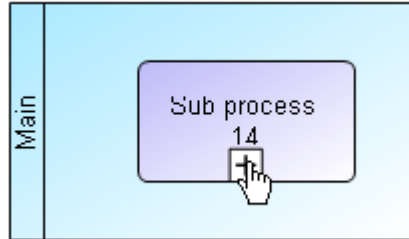
**Figure 68.** Link to independent process action



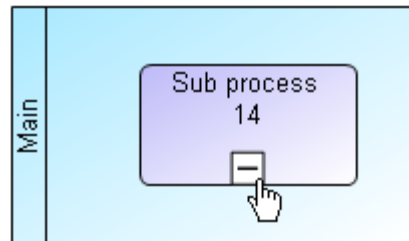
## Embedded (internal) sub processes

To create an embedded sub process click on the “+” to put the SubProcess into the expanded state. The sign will change to an “-” to indicate that if you click on it again the shape will resume its normal size and position.

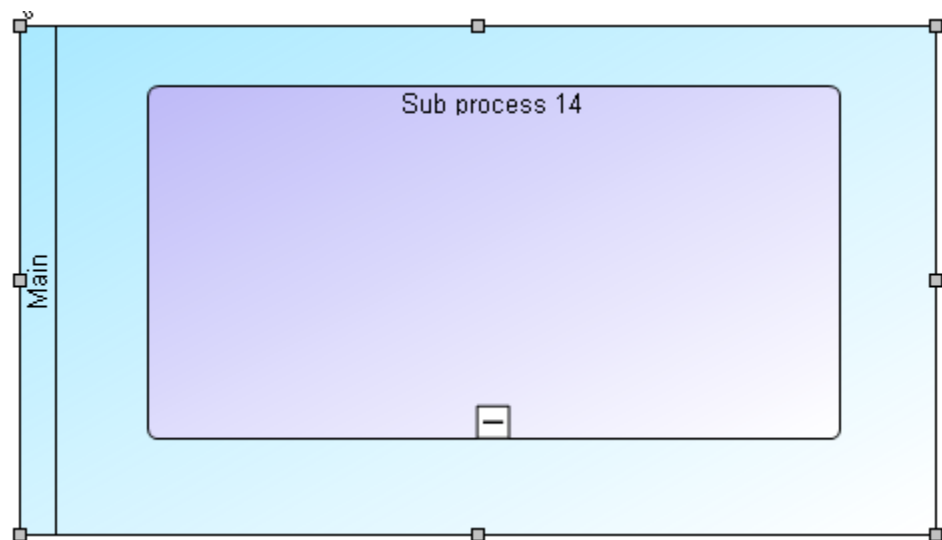
**Figure 69.** Shape showing un-expanded state



**Figure 70.** Shape showing expanded state

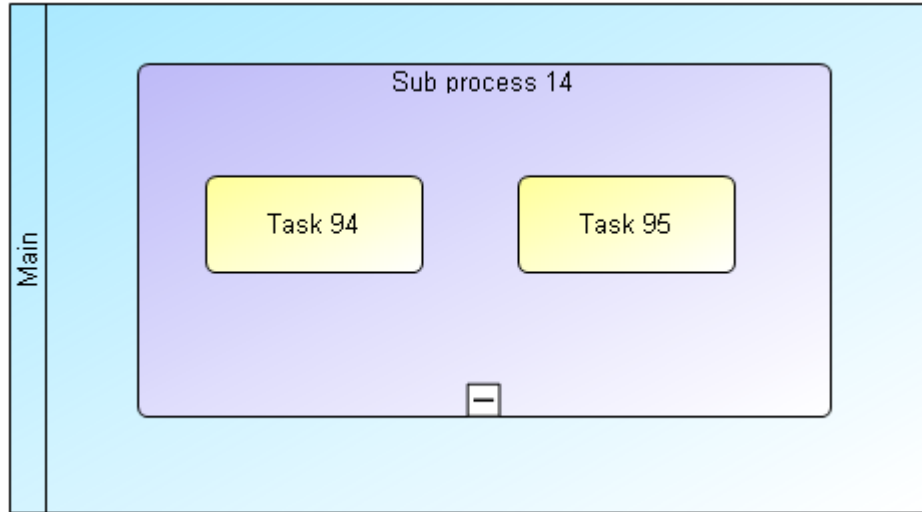
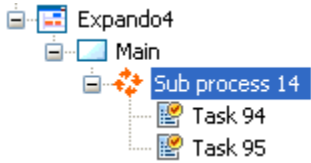


By way of example we will add some tasks in this subprocess. Firstly we will resize it to provide space. (This also means resizing the main pool).

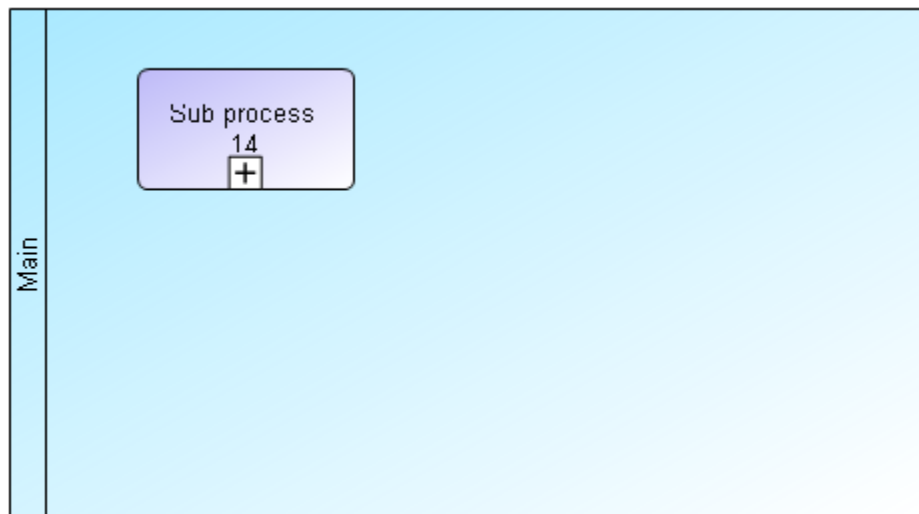




Next, we will add some tasks inside the expanded SubProcess using the BPMN toolbar. Make sure that you have the SubProcess *selected* when you do this, otherwise the task will not be created as children of the SubProcess.



Now let's un-expand the SubProcess by clicking on the “-” indicator.



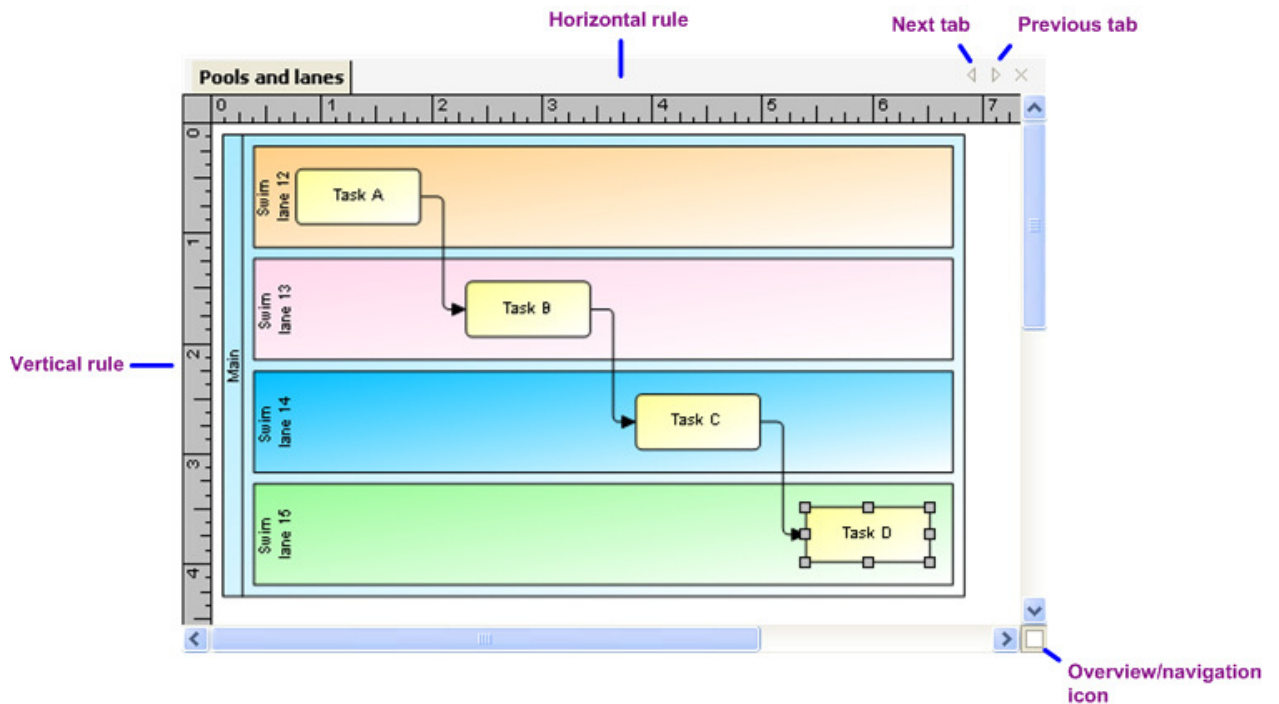
The SubProcess reverts back to its normal size and shape.

# The Diagram Editor

*This chapter describes the operation of the diagram editor. The diagram editor is used to visually edit Process Diagrams*

## Diagram editor windows

**Figure 71.** Introducing the diagram editor



If you have any experience of other diagramming tools then you should find learning to use Avantage quite simple because Avantage implements a very standard set of drawing functions.

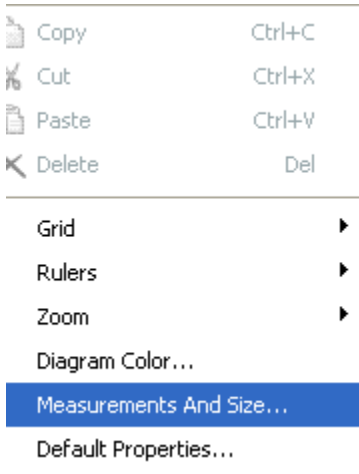
The major difference between Avantage and other diagramming software is that Avantage is model/object based rather than file based. Process diagrams are not saved to individual files in the file system, instead they are objects of the process model to which the diagram belongs. The data for all objects in a particular model are saved into a *model* file which is an XML object repository.

Another major difference is that Avantage has an intelligent layout manager. This can save large amounts of editing time and ensure that your diagrams are geometrically resonant.

## The Canvas

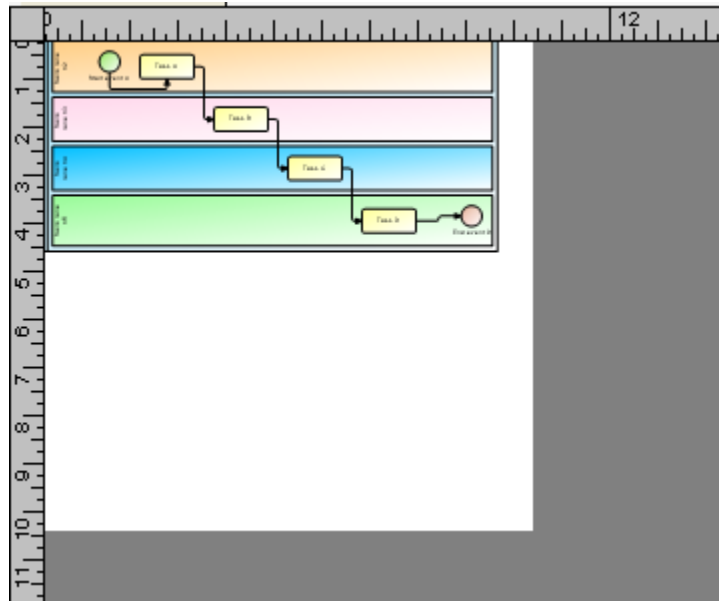
The Canvas is the surface on which you compose your process diagram. The size of your canvas determines how large your process diagram can be.

### To change the size of your canvas



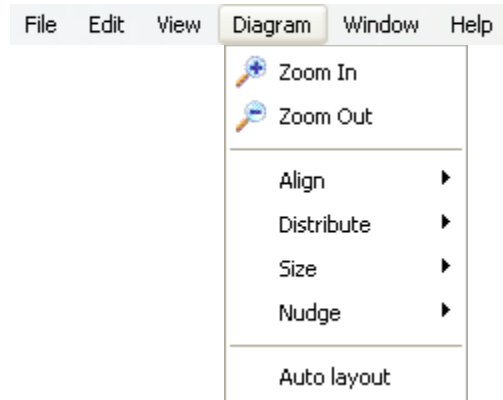
9. Right click on canvas surface
10. Choose Measurements and Size action
11. Use the Measurements and Size dialog to change the Canvas Height and or Canvas Width.
12. Click OK and process diagram is redrawn on the new Canvas.

**Figure 72.** A diagram zoomed out to show extents of canvas



## The Diagram Menu

The Diagram Menu is available whenever a process model diagram is open and is the active window in your workspace.



## Rulers

You can use the rulers to move shapes to a precise positions on the canvas. Rulers are calibrated in the current unit of measure specified in the canvas.

When you move a shape, a shadow appears on both the vertical and horizontal rulers to indicate the exact position of the shape. Precision at this time can save a lot of editing work later.

## Selections

You can select a single shape by clicking it. Most shapes can be selected by clicking anywhere in them. For Pool, Lane and SubProcess shapes you must click on their labels.

To select multiple shapes, hold the **CTRL** key down and then click each of the shapes that you want to include in the selection, or click and drag on empty space to draw a selection a marquee rectangle. The last shape selected for a multiple selection is distinguished from the others by its gray selection handles. The gray handles are used to indicate that it is an anchor shape.

Avantage uses anchor components as a reference or anchor for alignment operations. If you want another shape to serve as the anchor, hold the **CTRL** key and then click on another component within the selection. Avantage will transfer the gray handles from the previous shape to the selected component.

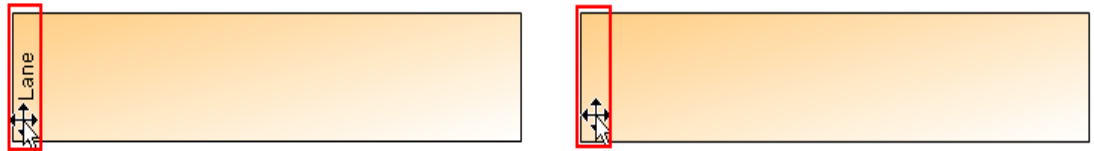
To deselect a single shape from a multiple selection, press the **SHIFT** key and click the shape that you want to exclude from the selection.

## Selecting Pools, Lanes and expanded SubProcess shapes

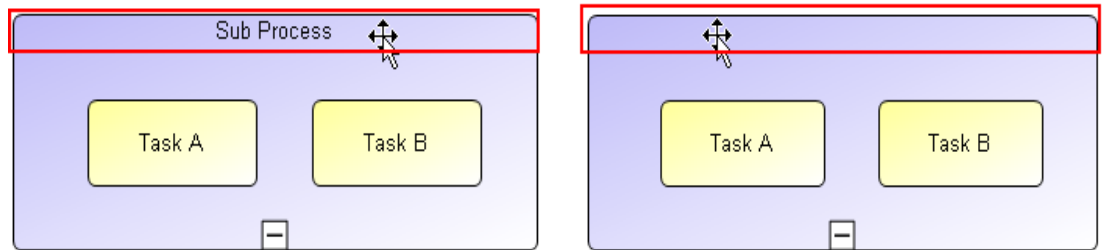
In order to select a container shape (i.e. a Pool, Lane or an expanded SubProcess) you must click on its Label or the place where the label would normally appear if the *Show Labels* property is false.

In the

**Figure 73.** Selecting a pool or lane

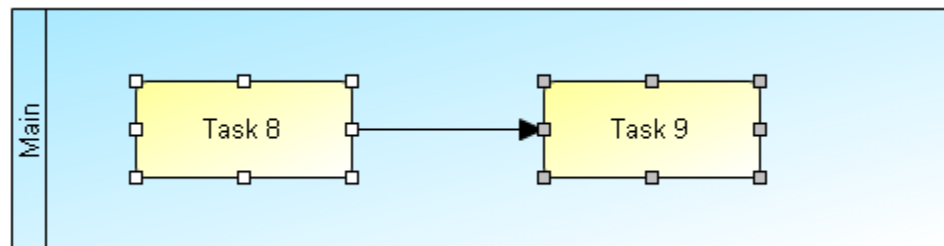


**Figure 74.** Selecting an expanded SubProcess



Clicking anywhere but on the selection sensitive area will de-select the shape.

**Figure 75.** Image showing anchor selection (grey handles)



## Moving things around



You can move selected shapes by clicking them and dragging the mouse. When you place the pointer over a shape that you can move, the mouse pointer changes to a can-move pointer.

*Note: To drag a pool, lane or expanded subprocess, you must click the mouse on the label.*

If the Snap to Grid option in the canvas settings is enabled, the top left corner of the shape's bounding box aligns with the grid.

You can move the selected shapes slightly using the nudge operations. These commands move the shapes one unit by default and five units if you are holding the `SHIFT` key.

Movement is ultimately constrained by a set of rules which examine the logical movements of shapes. Advantage will not allow you to drag a shape to where it does not belong in the process model.

An easy way to see the effect of a logical constraint is to attempt to drag a Task shape outside of its enclosing swimlane or pool.

## Changing the size of shapes

If a shape can be resized, the pointer turns into a directional arrow when you place the pointer over the one of the control handles on the shape.. Depending on which selection handle you select, you can either scale the object horizontally, vertically, or both.

If you hold the `SHIFT` key while you scale an object, the horizontal and vertical scale factors are always equal. If you hold the `CTRL` key while you scale a shape, the scaling performed is relative to the center of the shape.

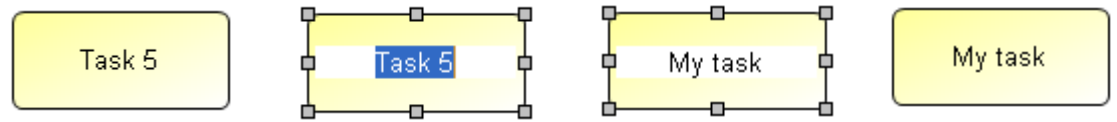
## Changing label text

Double-clicking on a shape's text will put the label into editing mode. Change the text and then click somewhere else to commit the changes.

You can also change a shape's label text by changing it in the Workspace Navigator.

By default, some shapes do not show any label text. Those shapes include events, sequence flows, message flows and associations. You can force a shape to show its label text by right-clicking on it and selecting the “Show label(s)” action.

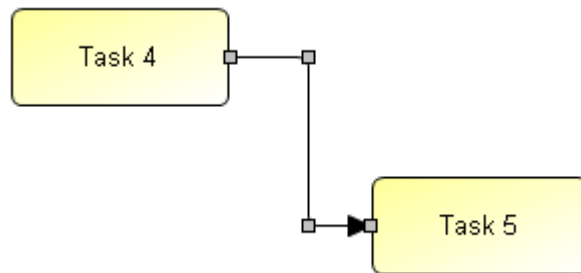
**Figure 76.** Steps in changing label text



## Editing sequence flow visual route

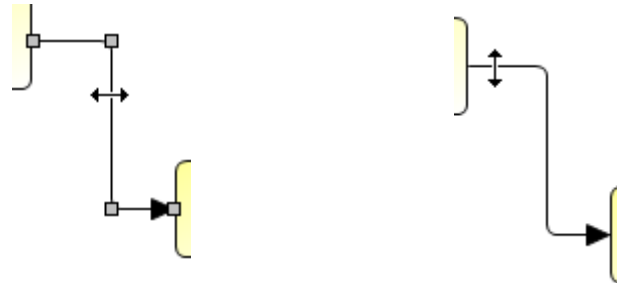
A sequence flow connection is an orthogonal line. Avantage will automatically re-arrange the routes of sequence flows when you move an object which it connects to.

A sequence flow can consist of any number right angle turns (vertices) on its route.



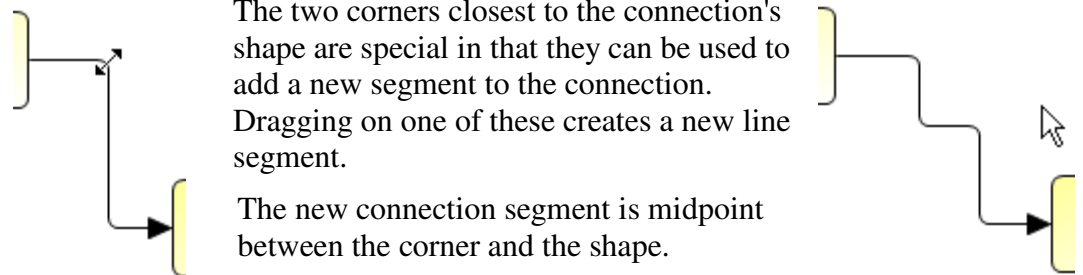
The selection is marked by gray boxes. Each gray box is either an end-point or a vertex.

## Changing the vertical or horizontal position of a line segment



The mouse cursor will indicate the operation applicable to the segment. Dragging in an allowed direction will relocate the link segment.

## Adding a turn (vertex) to a connection



The two corners closest to the connection's shape are special in that they can be used to add a new segment to the connection. Dragging on one of these creates a new line segment.

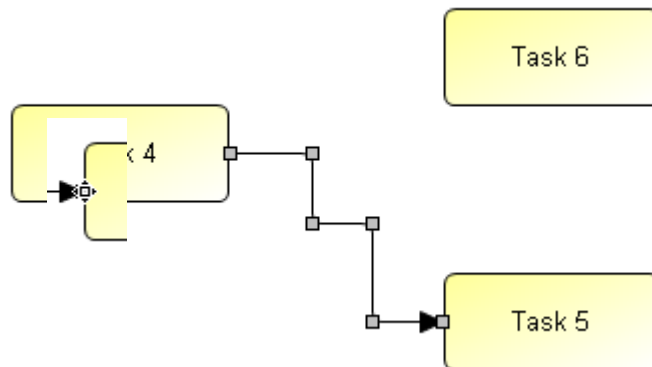
The new connection segment is midpoint between the corner and the shape.

## Re-linking connections

A connection end point can be relocated from one shape to another.



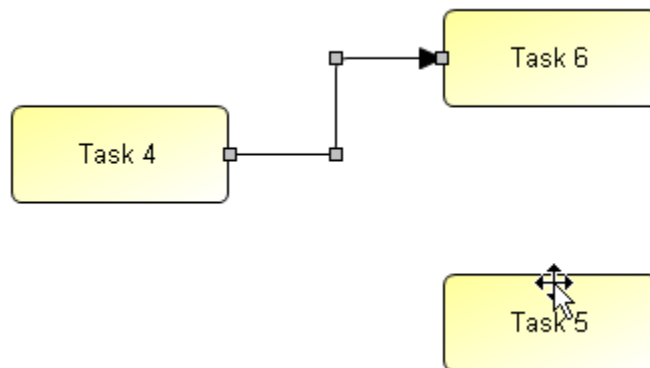
3. Select the connection



4. Position the cursor over the end-point to be relocated.  
The mouse cursor changes to a Can-Retarget indicator when you are position at the right spot.

5. Drag the mouse to the new target shape.

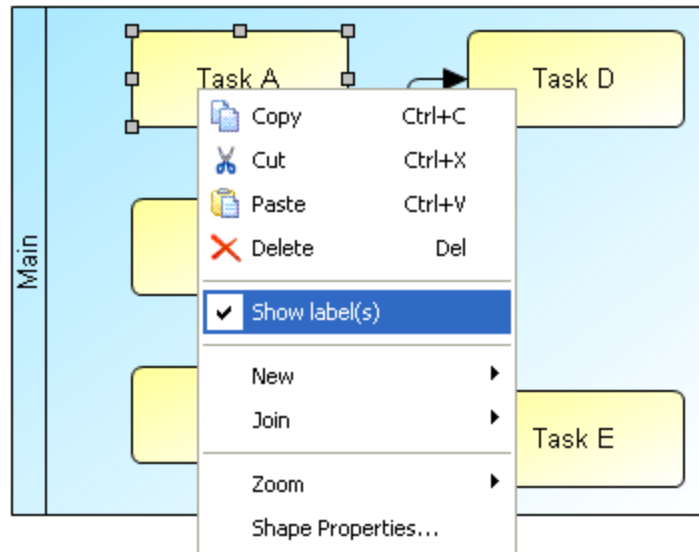
This process is identical to drawing a new link with the BPMN toolbar. Releasing the mouse will re-link the connection to the new target.



## Shape actions

Shapes have a set of actions associated with them. There is a standard set of clipboard actions (copy, cut, paste and delete) and a standard set of actions for changing a shape's visual properties. The shape may also have contribution menus from the type of process model object the shape represents.

**Figure 77.** Shape context menus



## Show labels

A shape's label text can be hidden or shown. By default most shapes show label text.

- *Note:* You probably noticed that the labels are moved together with the corresponding shapes. How can the labels be managed independently? Hold the **Ctrl and Shift** keys pressed until you select the label you want to move or scale.

## Shape properties...

Change visual attributes of a shape or selection of shapes (colour, line size, font etc).



## Auto layout

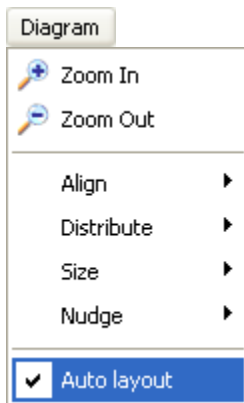
Avantage includes advanced auto layout functionality. This is the ability to automatically arrange the location and placement of tasks, pools and swimlanes and other flow objects according to an algorithmic set of rules.

Auto layout can save a lot of time and effort and allow you to concentrate on process modeling instead of placement of shapes.

Auto layout is a mode of diagram editing. By default it is not enabled.

Auto layout is invoked automatically by the diagram editor if you edit a .model file which has never before been edited and it does not contain any canvas coordinate data. For example, models created from a conversion of import action.

Auto layout is not a panacea, there are limitations to its usefulness but it can save you a lot of work.



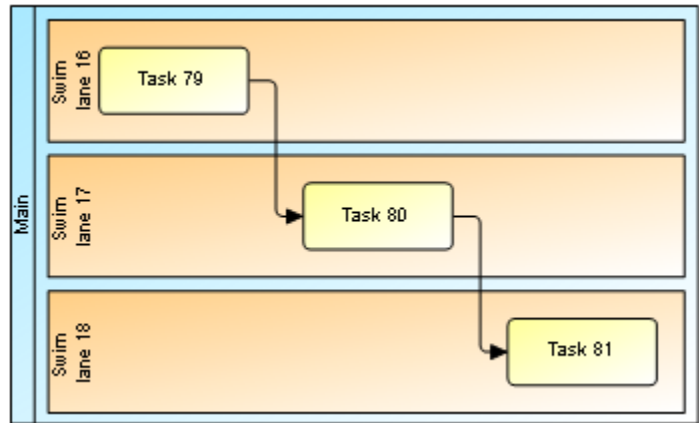
## Enabling Auto layout

To enable auto layout check the Auto Layout menu item in the Diagram menu ON. This menu is a checkbox menu. When it is checked (ticked) the feature is ON, when it is unchecked (not ticked) the feature is OFF.

When you switch the mode ON, the diagram is laid out according to the layout manager's rules.

## Forcing re-layout

At any time you can force a re-layout your BPMN diagram by clicking the "Force layout" button on the BPMN toolbar. This action works on the diagram regardless of whether auto layout is set as mode or not.

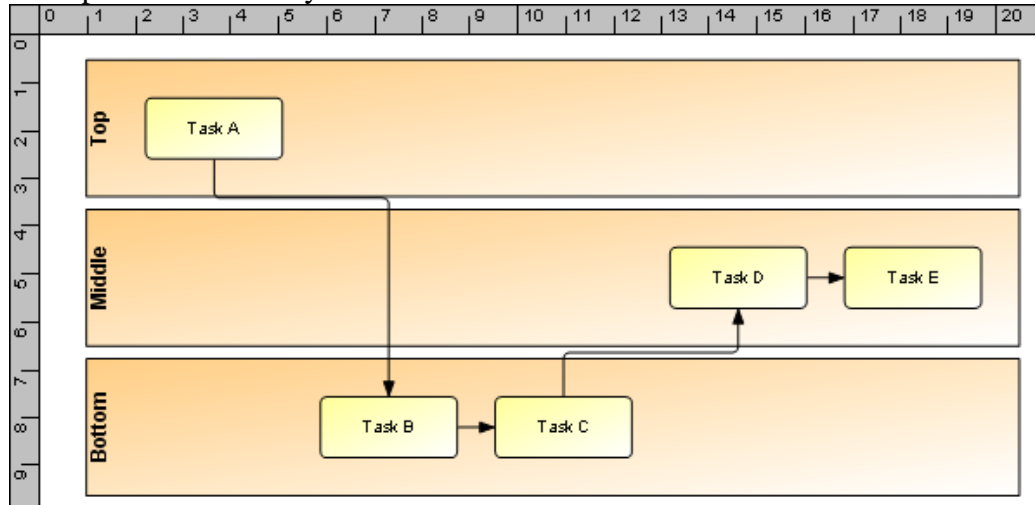


## Moving pools and lanes

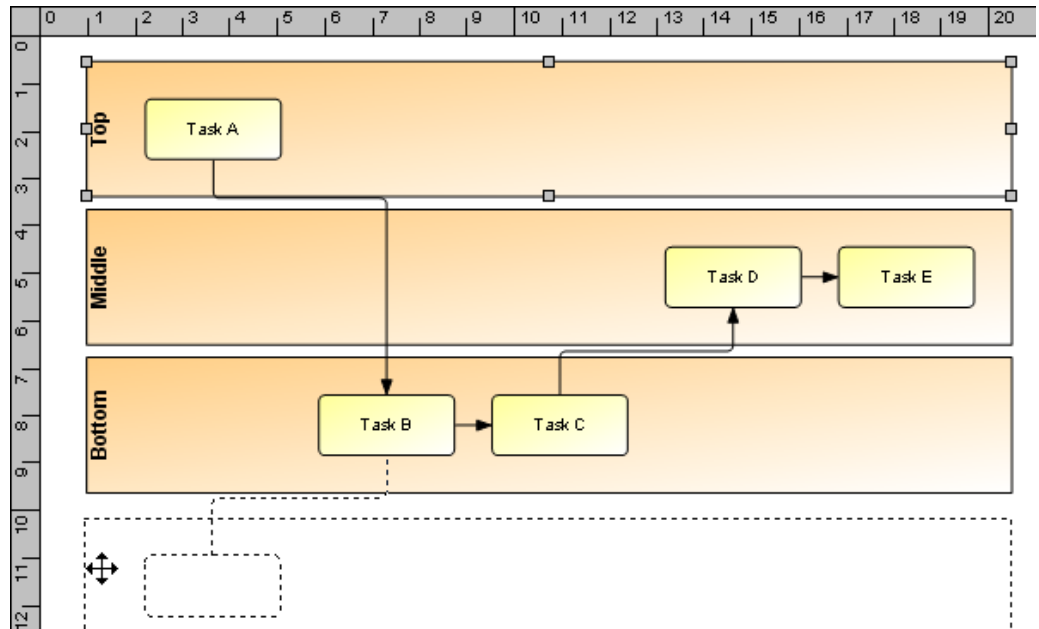
When auto layout is ON, you can easily relocate lanes and pools with the dragging gesture. Notice that when you drag a pool or lane a ghost marquee outline is drawn. The top line of the ghost marquee is the position in the vertical order.

To move a lane downwards, drag it so that the ghost marquee's top line is below the top line of the lane you want to move it under.

To move a lane upwards, drag it so that the ghost marquee's top line is above the top line of the lane you want to move it above.

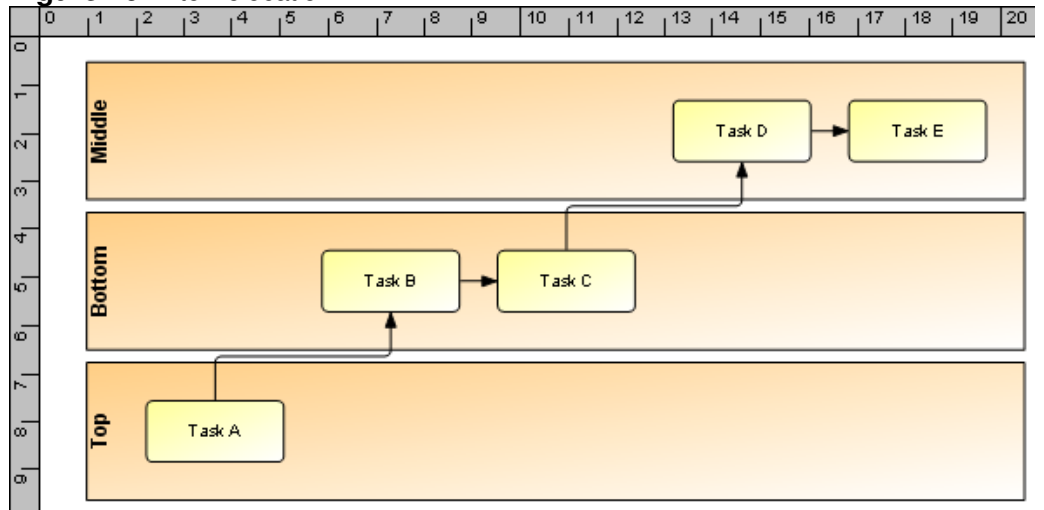


**Figure 78.** Dragging top lane to the bottom



After you release the mouse the layout editor will re-arrange the lanes using the new vertical ordering.

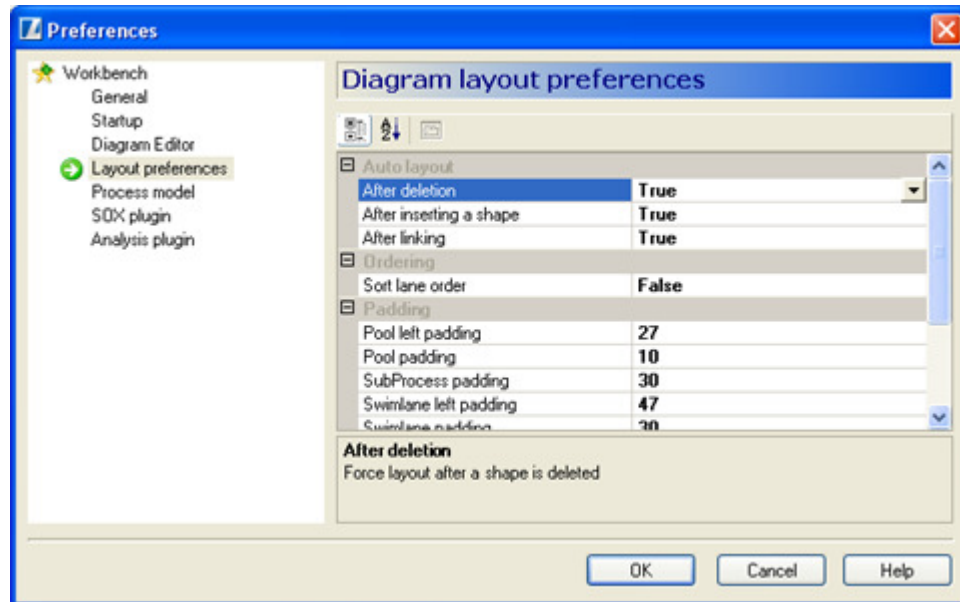
**Figure 79.** After relocation



## Layout Preferences

It is possible to customize the behaviour of the layout manager by changing settings here.

**Figure 80.** The layout preferences page



### Auto layout triggers

You can specify which editing actions trigger auto layout of the diagram.

#### After deletion

Layout takes place immediately after you delete a shape or group of shapes from the diagram.

#### After inserting a shape

Layout takes place immediately after you insert a shape or shapes into the diagram.

#### After linking

Layout takes place immediately after you connect two objects with connecting link of some kind or after a link has been re-routed from one shape to another.

### Lane ordering

This setting controls the vertical ordering of pools and lanes. They can be in canvas order or sorted order. By default they are in canvas order.

### Sort lane order

When **True** this setting causes the vertical ordering of pools and lanes to be determined by the auto layout manager itself.

When **False**, the vertical ordering of pools and lanes is determined from their existing positions in the Canvas. By default this setting is **False** in order to allow you to have control of vertical placement.

## Padding

### Pool left padding

This is the amount of blank space between the left edge of a pool before the first contained object (like an activity or lane) can appear.

### Pool padding

The amount of blank space around the inside edges of a pool object.

### SubProcess padding

The amount of blank space around the inside edges of a subProcess object.

### Swimlane padding

The amount of blank space around the inside edges of a swimlane object.

### Swimlane left padding

The amount of blank space between the left edge of the lane and any objects it contains.

## Sizes

### Activity horizontal space

### Activity vertical space

### Minimum expanded SubProcess size

The minimum size of a subprocess object in terms of width and height.

### Minimum pool size

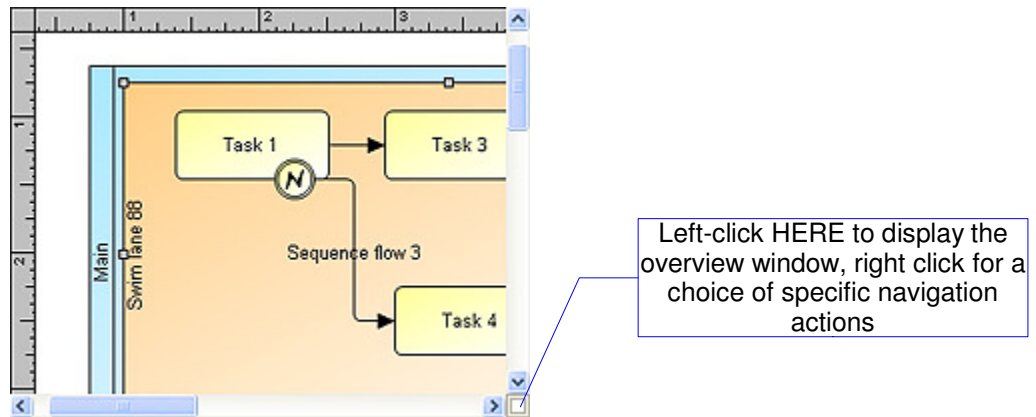
The minimum size that the layout manager will reduce a pool object to in terms of width and height.



## Navigating within a large process diagram

When a process diagram becomes wider and or taller than the window it is being displayed in the scroll bars get turned on. Yes, we all know this and it necessitates us having to operate the scroll bars, or change the zooming.

**Figure 81.** The overview icon



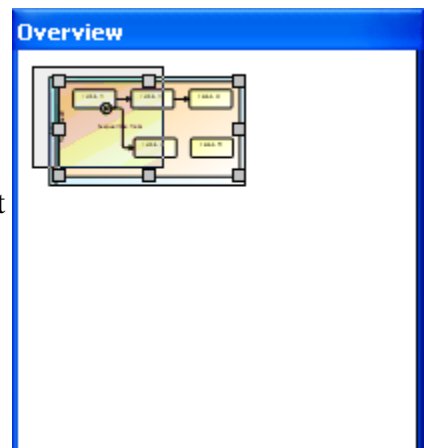
## Using the Overview window

**Figure 82.** Overview window

The overview window pops up when you left-click on the navigate icon as shown above.

A shaded rectangle is displayed to indicate just where in diagram you are currently at.

While holding the left mouse button down, drag the rectangle around within the preview window. As you drag the rectangle around the main diagram window tracks it.



## The Navigate actions

When you right-click on the overview icon you are shown a set of specific navigate actions

### Left edge

Scrolls to the left most edge of the diagram.

### Right edge

Scrolls to the right most edge of the diagram.

### Page left

Scrolls left one page at a time

### Page right

Scrolls right one page at a time

Scroll Here
Left Edge Right Edge
Page Left Page Right
Scroll Left Scroll Right

## The BPMN toolbar

The BPMN toolbar is displayed whenever a diagram editor window is open. It is used for inserting BPMN objects into a diagram.

### How to use the BPMN toolbar

Click on a tool icon in order to select and arm the action and then move the mouse to the location on the canvas where you want to insert the object. As you move the mouse over the canvas the mouse cursor will indicate whether the operation is allowed or not at that location in the canvas. Click the mouse when you have moved it to where you want the shape to be inserted.

*Note:* The procedure for inserting connection objects is slightly more involved. Please read the “Linking tools” section for details.

**Figure 83.** The BPMN toolbar



### Screen snapshot tool

This tool creates an image file of the diagram. The image formats supported are *.bmp*, *.jpg* and *.emf*. When you click the button you are prompted for a location in which to save the image.



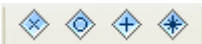
### Activity tools

#### Add Task

Adds a Task object

#### Add SubProcess

Adds a subprocess object.

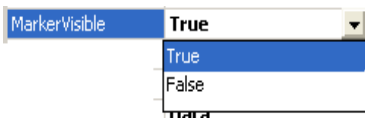


### Gateway tools

#### Add XOR gateway

Adds an XOR gateway object.

**Figure 84.** The default XOR shape is a simple diamond



In order to display the alternative “X” graphical indicator you must set **MarkerVisible=True** using the Properties inspector.

**Figure 85.** XOR shape showing “X” indicator



### Add OR gateway

Adds an OR gateway object.



### Add AND gateway

Adds an AND gateway object.



### Add Complex gateway

Adds a Complex gateway object.



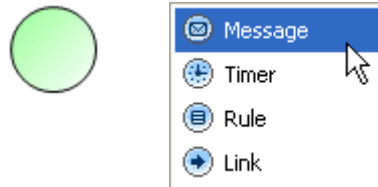
## Event tools

These tools create event objects. Events have triggers or produce results. Event objects created with these tools do not have any trigger defined. You can add triggers with the freshly created event object's menu contributions.

By default, newly inserted event shapes do not show a graphical indicator. This is because the new event does not yet have any defined Triggers (in the case of End events Results).

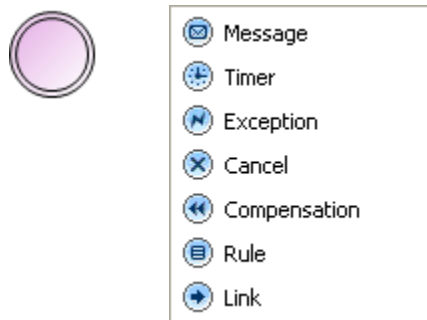
### Add Start event

Adds a Start event object.



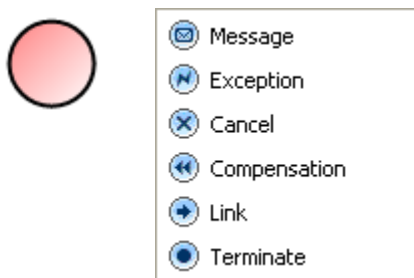
### Add Intermediate event

Adds an Intermediate event object.



### Add End event

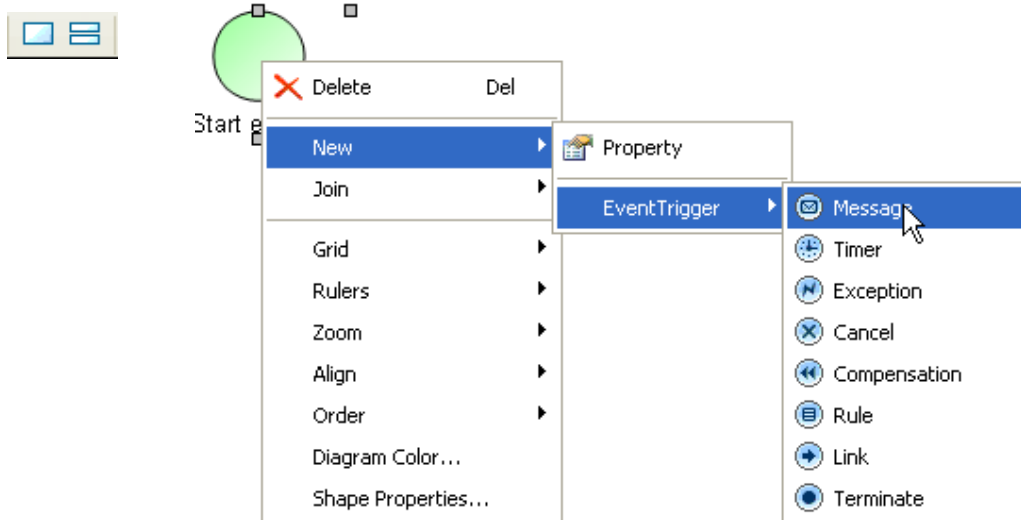
Adds an End event object.



## Adding an event trigger to an event

To add a trigger to a freshly inserted event object right-click on the event and choose the New EventTrigger action. Choose from one of the listed triggers.

**Figure 86.** Adding a trigger



- *Note:* In order to make a BPMN *Multiple* event you must assign more than one trigger to the event.

## Swimlane tools

### Add Pool

Adds a Pool to the diagram.

- *Note:* The main pool can be visible or invisible, as determined by the **BoundaryVisible** property. Use the Property Inspector to change the **BoundaryVisible** property. There can only be one invisible pool in a diagram.

You can globally set this property in the Process Model Preferences page option “**Default main pool is invisible**”.

### Add Lane

Adds a Lane to the selected pool.



## Artifact tools

### Data object artifact

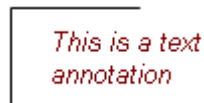
This action inserts a data object artifact.



Artifact 2

### Text annotation artifact

This action inserts a text annotation. To edit its text, double-click on it.



### Displaying document information

Text annotations can also be used to display the basic BPMN Document Properties of a Diagram. You do this by by setting a format string into the **Template** property of the artifact instead of editing its label.

*Note: In order to change the Template property you will need to use the Properties inspector.*

In order to display a property you need to place a ^ character and then a letter which identifies the property. For example the template string :

“Diagram: ^N Created by: ^A Version: ^V”

will appear in the annotation as

“Diagram: **Diagram 16** Created by: **Rebecca\_134** Version: **1.43**”

To display a literal ^ character you must enter ^^

As soon as you enter any text at all into the **Template** the **Text** property is ignored and instead the **Template** provides the text for the annotation. To disable the effect of a template you must clear the **Template** property to an empty string.

On the next page is a table of the command letters used to display the various document properties

*Note: The alphabetic case of the command letter is important*

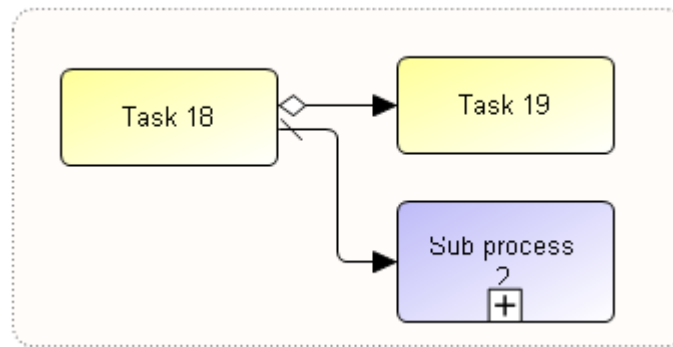
<b>Command letter</b>	<b>Command</b>	<b>Description</b>
A	Author	Name of user who created the diagram. (BPMN <b>Author</b> property)
C	CreationDate	The date the diagram was created. (BPMN <b>CreationDate</b> property)
c	CreationDate	ShortDate format of <b>C</b>
D	DiagramDescription	A description of the of the diagram. (BPMN <b>Description</b> property)
P	CanonicalPath	The <b>CanonicalName</b> of the diagram.
M	Modification date	The date that the diagram was last saved. (BPMN <b>ModificationDate</b> ) property
m	Modification date	ShortDate format of <b>M</b>
L	Insert line in text	A newline character is inserted in the text which will cause the text to wrap in the annotation.
N	Name	The <b>Name</b> of the diagram (BPMN <b>Name</b> ) property.
V	Version	The BPMN <b>Version</b> property of the diagram. The Version of the diagram is automatically incremented each time the document is saved. It is in the form of Version_Number.Revision_Number.  <i>Note: Only the revision number is automatically incremented. In order to change the Version number you should modify it directly with the Properties Inspector.</i>

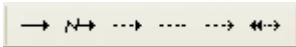


## Grouping artifact

This action inserts a grouping rectangle into the diagram. Drag and resize the grouping rectangle to enclose the group of BPMN shapes that you want to group.

*Note:* A grouping artifact has no active effect on the process model, it is a modeling convenience that is useful for process analysis. You can also use it as a property container for a group of shapes by adding properties to it.





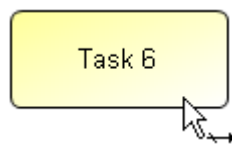
## Linking tools

Clicking on a linking tool arms the diagram editor to link between two flow objects.

To start the linking process, left-click the mouse in the source object and then whilst keeping the mouse button down, drag into the destination object.

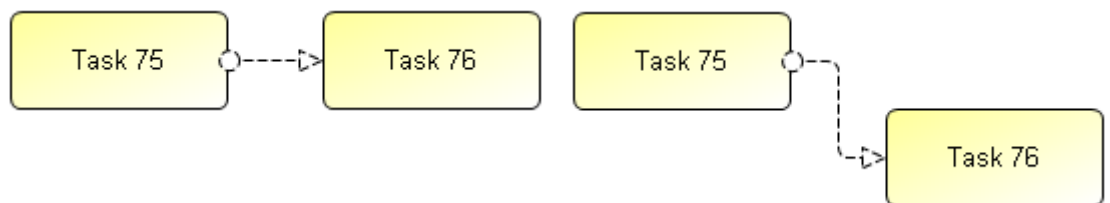
Release the mouse to complete the connection.

**Figure 87.** An armed mouse cursor over a linkable shape



## Add Message flow

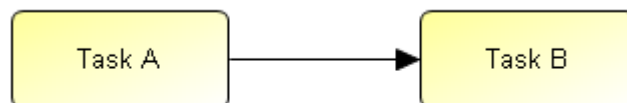
Adds a message flow connection between two flow objects in a diagram.

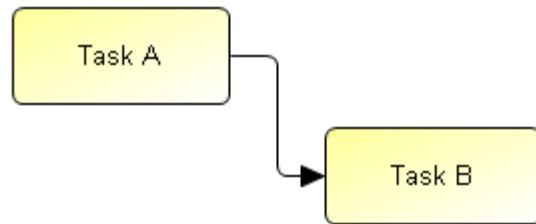


## Add Sequence flow

Adds a Sequence flow connection to the diagram. A sequence flow is an orthogonal connecting line.

**Figure 88.** Sequence flow links are orthogonal

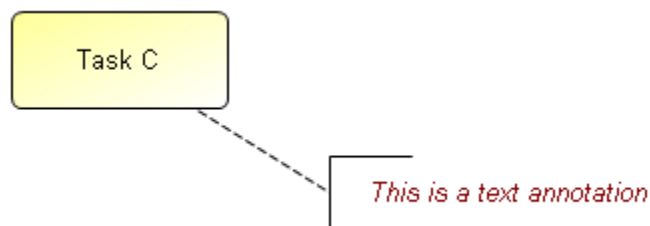




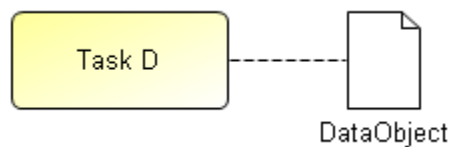
## Add Association

Adds an Association between a flow object and an artifact object. This is commonly used to attach text annotations and to specify the transacted data entities of a workflow.

**Figure 89.** Task connected to a text annotation with an association



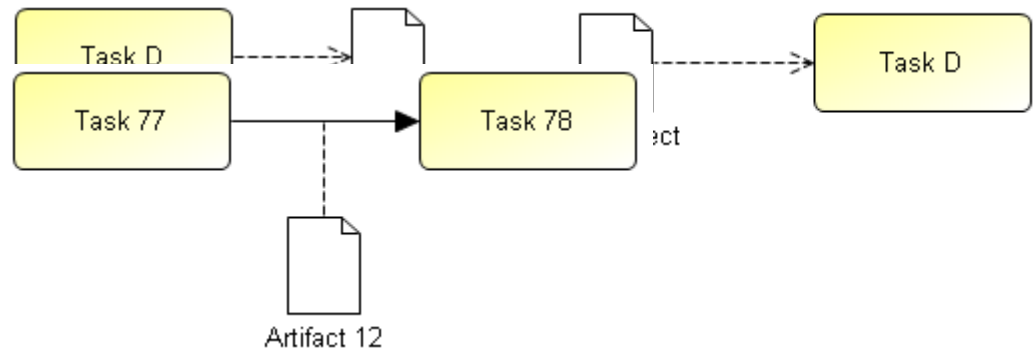
**Figure 90.** Task connected to data object artifact with association



## Add directional association

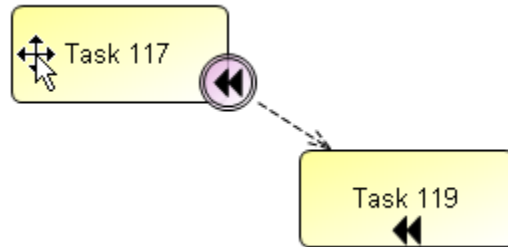
This toolbar button creates a directional association. The direction is determined by the direction of your drag gesture.

- *Note:* You can also set or change the direction of an association with the Property Inspector change the **Direction** property to any of **To**, **From**, **Both** or **None**.





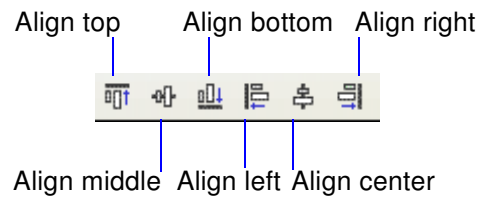
## Add compensation association



## Alignment tools

Contains actions for aligning shapes with respect to a given anchor shape.

### Alignment actions



### Align top

Horizontally aligns the selected shapes with the top of the anchor component.

### Align bottom

Horizontally aligns the selected shapes with the bottom of the anchor component

### Align middle

Horizontally aligns the selected shapes with the center of the anchor component.

## Align left

Vertically aligns the selected shapes with the left edge of the anchor component

## Align right

Vertically aligns the selected shapes with the right edge of the anchor component

## Align center

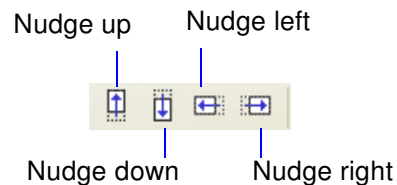
Vertically aligns the selected shapes with the center of the anchor component.

## Nudge tools

Contains actions for moving the selected shapes by one logical unit in any direction.

**NOTE:** You can also operate the nudge commands with the keyboard arrow keys (left, right, up and down).

**Figure 91.** Nudge actions



## Nudge right

Move the selected shapes one logical unit right.

## Nudge left

Move the selected shapes one logical unit left

## Nudge up

Move the selected shapes one logical unit up.

## Nudge down

Move the selected shapes one logical unit down.

## The sizing tools

**Figure 92.** Sizing actions

Make same height      Make same size



Make same width

## Make same height

Change the height of the selected components to match the anchor component

## Make same width

Change the width of the selected components to match the anchor component.

## Make same size

Change the width and height of the selected components to match the size of the anchor component.

## The Space equalization tools

Contains actions for arranging shapes with respect to each other. The spacing commands take *three* or more shapes and evenly space them between the minimum and maximum points in the range selected.

**Figure 93.** Space equalization actions

Equalize horizontal spaces



Equalize vertical spaces

## Equalize horizontal spaces

Space the selected components evenly between the left-most and right-most shapes selected.

## Equalize vertical spaces

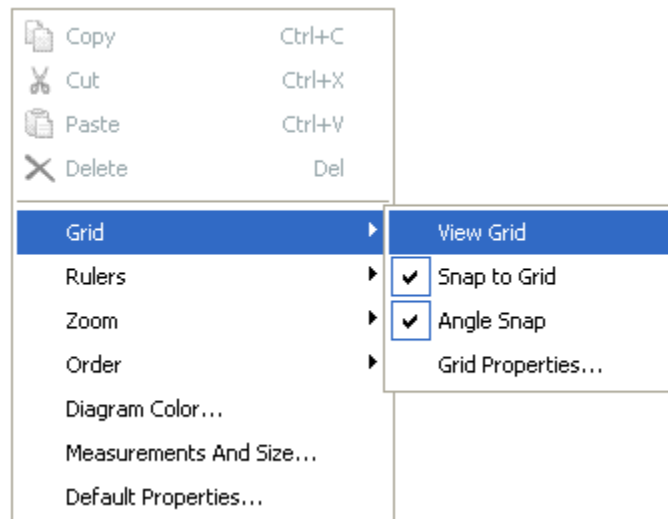
Space the selected components evenly between the top-most and bottom-most shapes selected



## Canvas level context menus

When you right-click the mouse on the drawing canvas a popup menu will appear that contains various actions you can do in order to configure your diagram.

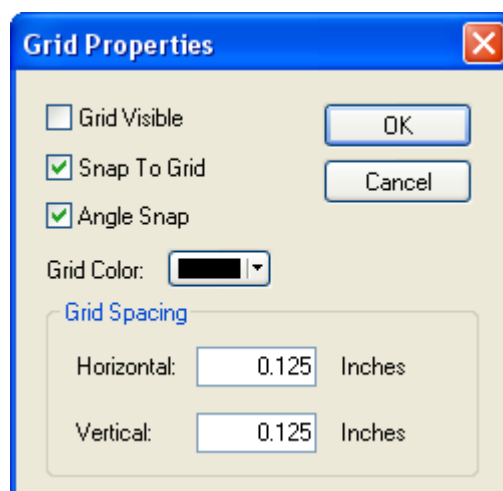
### Grid



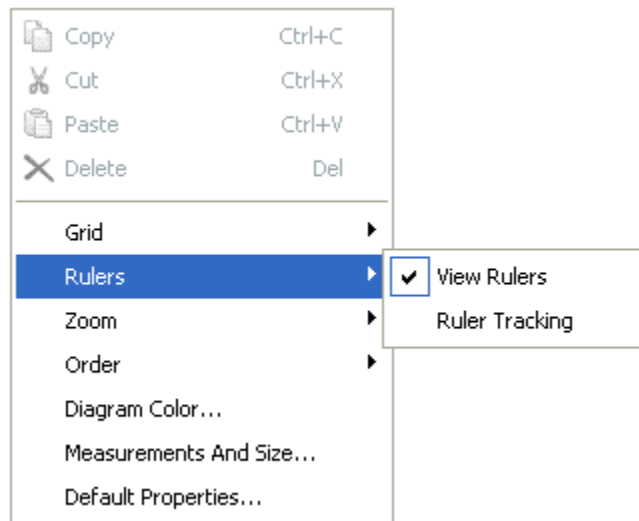
The grid is a set of evenly spaced points on the canvas that you can use to align shapes. When the snap-to-grid option is enabled, the top left corner of a shape is aligned to a grid point when the shape is moved.

The angle snap option limits rotation of shapes to fifteen-degree increments.

### Grid properties



## Rulers



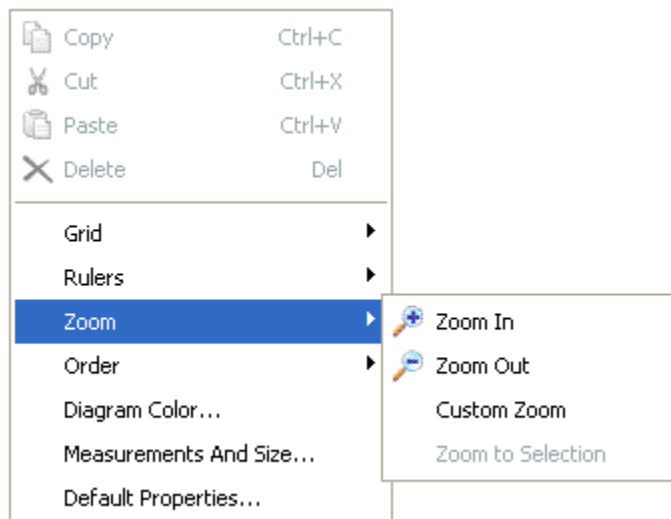
### View rulers

Toggles the visibility state of the Rulers. Hides and shows the rulers.

### Ruler Tracking

When ON, a shadow is drawn in the vertical and horizontal rulers to indicate the exact position of the shape on the canvas.

## Zoom



### Zoom In

Zooms the canvas in by 15 percent

## Zoom Out

Zooms the canvas out by 15 percent

## Custom Zoom

You can choose the zoom through a custom zoom dialog.

## Zoom to Fit

Automatically zooms the canvas to best fit the contents

## Zoom to Selection

Zooms the canvas centering on the selection.

## Diagram Color

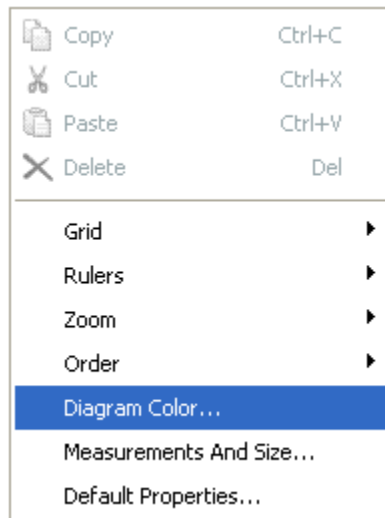
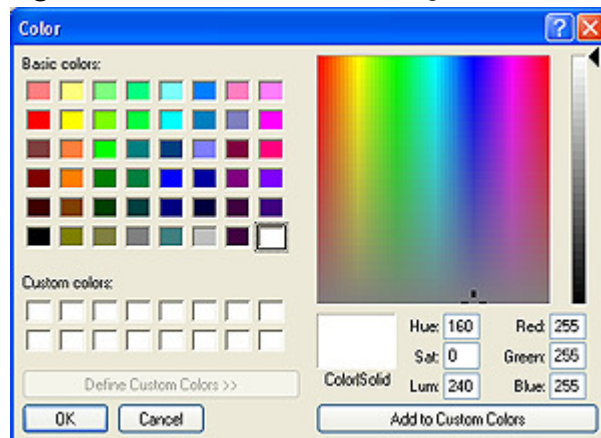
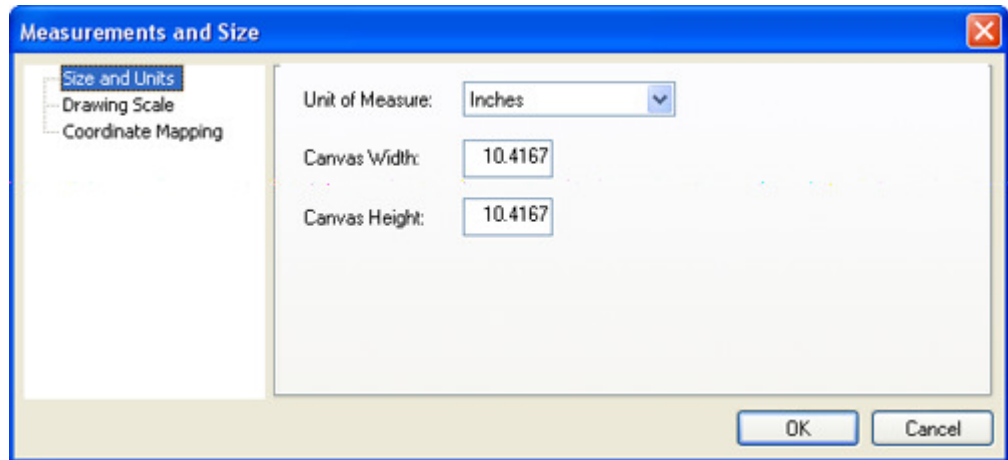


Figure 94. The color chooser dialog



## Measurements and Size

The Measurements and Size action dialog provides a way for you to specify the units of measure and drawing scale of the drawing canvas.



### How to increase the size of your process map

The *Canvas Width* and *Canvas Height* settings in the **Size and Units** options control the size of your process map. Enter the new dimension and then click OK.

You can use the Overview window to navigate around a large canvas. In the above screen shot, the overview is positioned at the bottommost and rightmost position of the canvas.

**Figure 95.** Drawing scale options

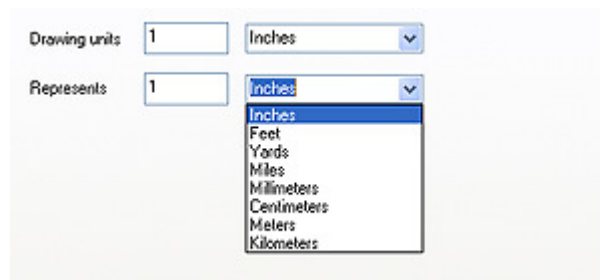
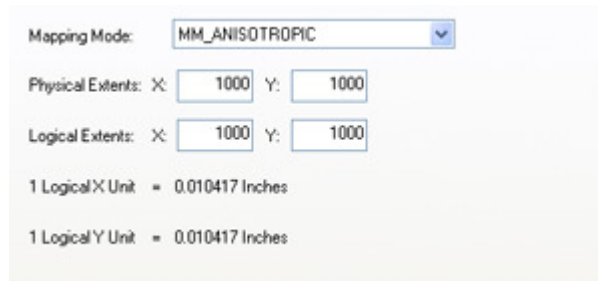


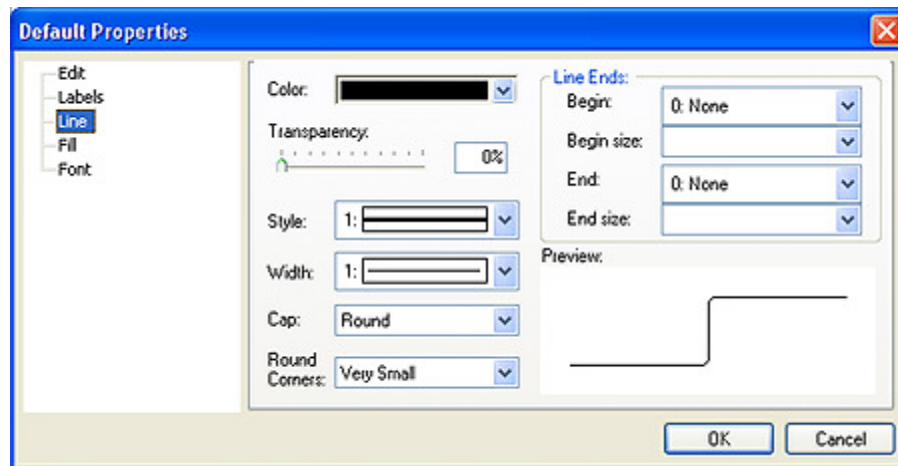
Figure 96. Coordinate mapping options



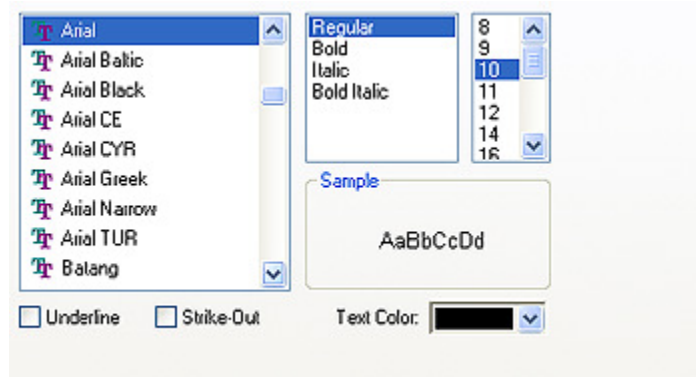
- *Note:* It is not recommended that you change any of the coordinate mapping options unless specifically instructed to do so.

## Default properties

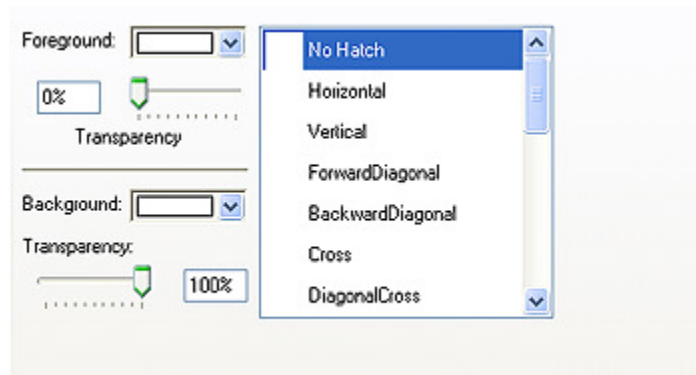
### Line properties



## Font selection



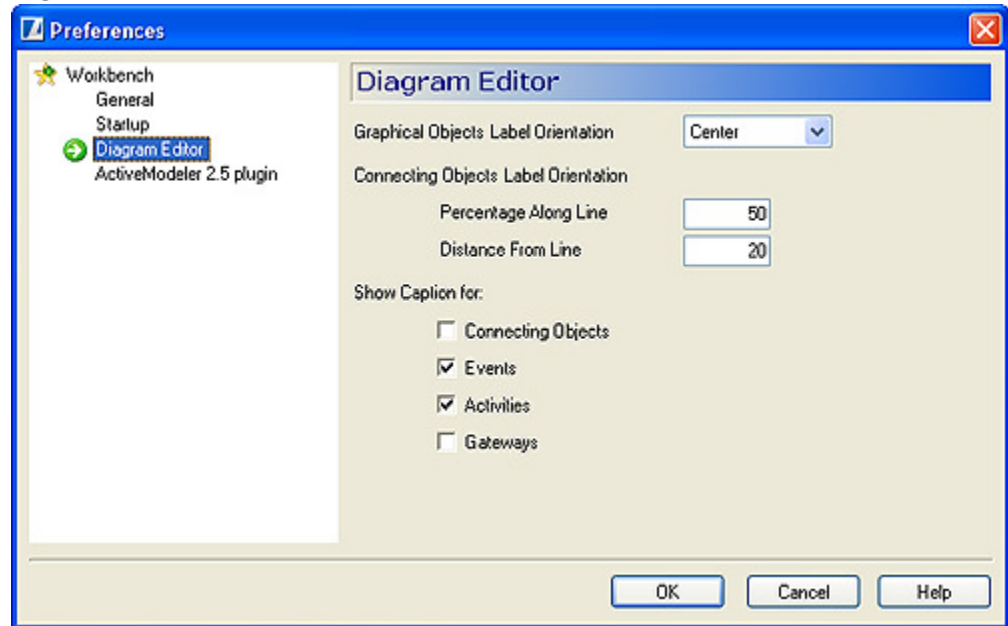
## Fill options



## Preferences Settings

The global preference settings for the diagram editor as displayed by choosing the [File:Preferences](#) menu action from the main menu bar.

**Figure 97.** The Diagram Editor preferences page



## Graphical Objects Label Orientation

This setting displays how the label text is oriented within the rectangle it occupies.

**Figure 98.** Orientations

- Top-Left
- Top-Center
- Top-Right
- Center-Right
- Bottom-Right
- Bottom-Center
- Bottom-Left
- Center-Left
- Center

## Connecting Objects Label Orientation

### Percentage along line

The default value of 50% centers the label along the line.

### Distance from line

The default value of 0 centers the label vertically with respect to the line. To move a label up enter a positive number (i.e 10). To move the label below the line enter a negative value (i.e -10).

### Show Caption for:

You can choose to display or not display captions for the four broad categories of shapes in a BPMN diagram

Connecting objects (SequenceFlow, MessageFlow, Association), Events, Activities (Tasks, SubProcess, Process) and Gateways.

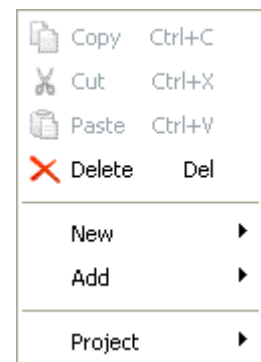


# Process Navigator Actions

*This chapter describes the built in process model actions in Avantage that you can access with context menus from the Navigator and Diagram editor windows.*

## Common Context Menu Actions

When you right-click the mouse on any item in the Navigator tree you will see a context menu. The exact details that you see in the context menu will depend on the kind of object that you have clicked on.



### Copy

This action copies the selected object(s) to the clipboard. If an object cannot be copied this action is disabled.

### Cut

This action copies the selected object(s) to the clipboard and then deletes them from the process model. If the object cannot be deleted or copied the action is disabled.

## Paste

This action pastes the clipboard contents into the process tree as a child of the selected object. If the type of object in the clipboard is an inappropriate child for the selected object the action is disabled.

## Delete

This action deletes the selected object(s) in the tree. This action is disabled if the object cannot be deleted.

## New

This item is displayed on objects which may contain children (for example diagram objects, flow objects, pools and lanes). This action is used to create new flow objects.

## Project

This item displays a list of actions that pertain to adding and removing projects from your Workspace.

## Add

This item displays a list of actions that pertain to adding items to projects.

## Renaming items

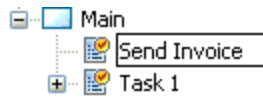
1. Select the item to be renamed by clicking on it once.



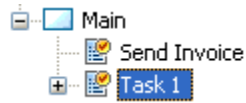
2. Click on it again. This puts the label into edit mode. The mouse cursor changes to a caret (insertion point).



3. Enter the new label text



4. Click anywhere outside of the label box to commit your changes to the model.

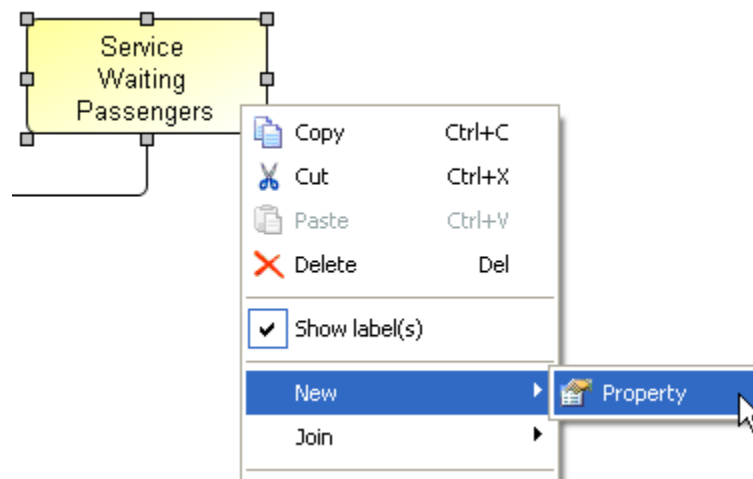


## Task object actions

<ul style="list-style-type: none"> <li> Copy    Ctrl+C</li> <li> Cut    Ctrl+X</li> <li> Paste    Ctrl+V</li> <li> Delete    Del</li> </ul>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">  Property         </div>	<ul style="list-style-type: none"> <li> Message flow</li> <li> Sequence flow</li> <li> Association</li> <li> Compensation association</li> <li> Exception flow</li> </ul>						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">New</td> <td style="text-align: right; padding: 5px;">▶</td> </tr> <tr> <td style="padding: 5px;">Join</td> <td style="text-align: right; padding: 5px;">▶</td> </tr> <tr> <td style="padding: 5px;">Project</td> <td style="text-align: right; padding: 5px;">▶</td> </tr> </table>			New	▶	Join	▶	Project	▶
New	▶							
Join	▶							
Project	▶							

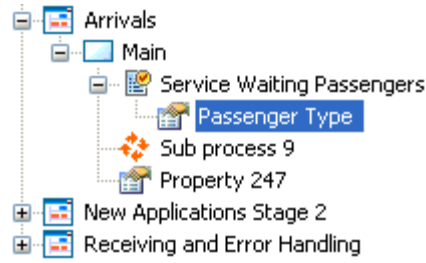
## New Property

Creates a new Property object. Use this action to add custom properties to objects.



In this case a new property is added to the “Service Waiting Passengers” activity and now belongs to the process model.

The scope of the property here is just at the “Service Waiting Passengers” level as it was added at the activity level.



The property has been called “Passenger Type”. Notice the “Property 247” defined at the Main pool level.

Looking at the Properties Browser the property type can be set as follows:

CanonicalName	Arrivals.Main.Service Waiting Passen
Correlation	<b>False</b>
Description	
Documentation	
ID	<b>sledc284ccdf54ee7971233f72</b>
IsXPDLDataField	<b>False</b>
Name	<b>Passenger Type</b>
ParentID	<b>ta788ed27878f64a31a8cc7e36</b>
Type	<b>String</b>
Value	<ul style="list-style-type: none"> <li>Byte</li> <li>Int16</li> <li>UInt16</li> <li>Int32</li> <li>UInt32</li> <li>Int64</li> <li>UInt64</li> <li>Single</li> <li>Double</li> </ul>

Such properties are now saved as part of the process model and can be used with the Lua script engine embedded in Avantage to produce powerful extensions to the Avantage program, e.g custom reports and analysis.

## **Join Message flow**

Joins the Task by message flow connection to another flow object.

## **Join Sequence flow**

Joins the Task by Sequence flow connection to another flow object.

## **Join Association**

Joins the Task by Association connection to another object.

## **Join Compensation association**

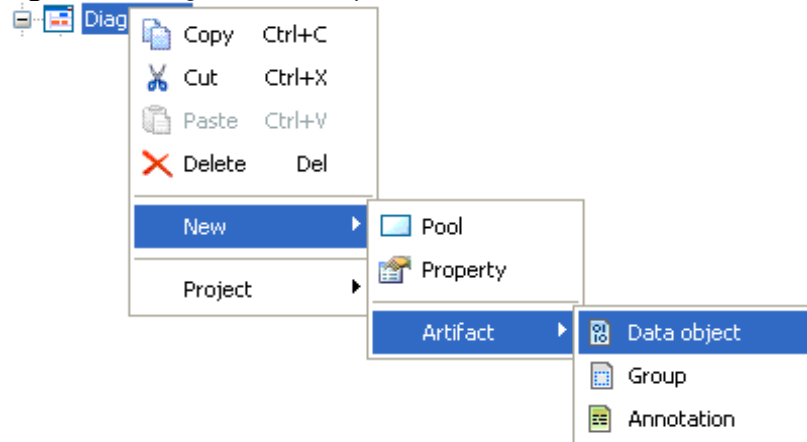
Joins the Task by a Compensation Association to another flow object.

## **Join Exception flow**

Joins the Task by an Exception flow to another flow object.

## Diagram object actions

Figure 99. Diagram menu expanded



### New Pool

Creates a new Pool object

### New Property

Creates a new Property object. The default type of the property is **string**.

### New Data object Artifact

Creates a new Data Object artifact.

### New Group artifact

Creates a new grouping artifact.

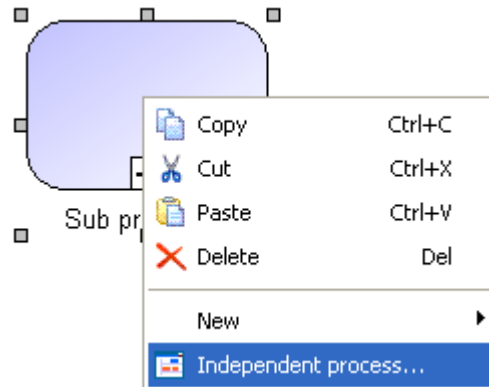
### New Annotation artifact

Creates a new text annotation.

## SubProcess object actions

### Independent SubProcess

**Figure 100.**Independent Process menu action



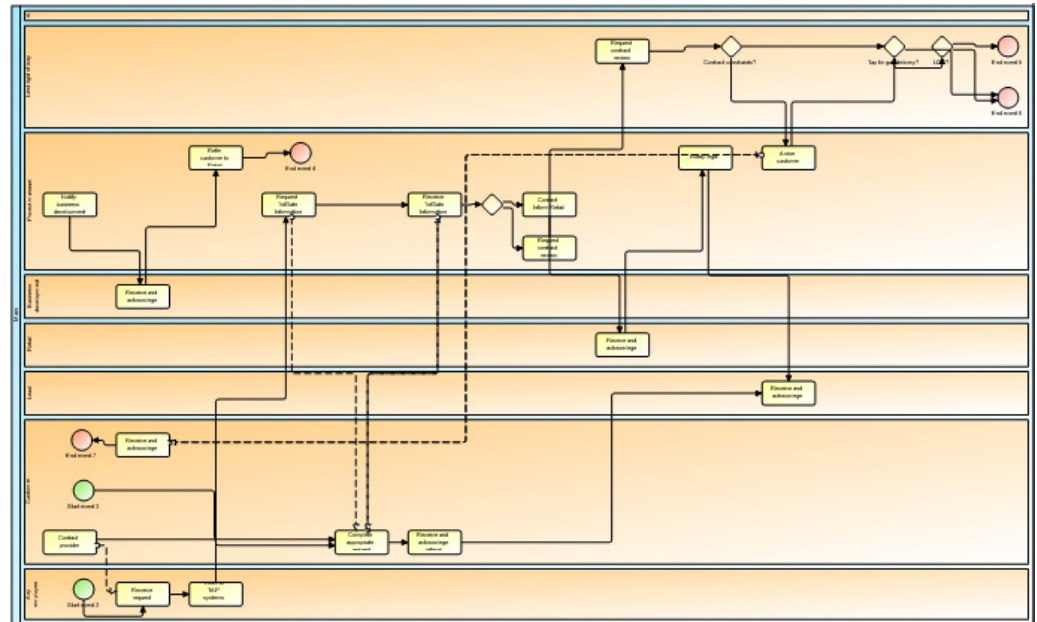
An Independent Sub-Process object is an activity within a Process that “calls” to another Process that exists within a BDP (see Figure 12). The Process that is called is not dependent on the Independent Sub-Process’ parent Process for instantiation or global data.

The called Process will be instantiated when called, but it can be instantiated by other Independent Sub-Process objects (in other diagrams) or by a message from an external source. The Independent Sub-Process object may pass data to/from the called Process.

The called Process will exist in a separate diagram that can have multiple Pools. Any view of the called Process (including an expanded view within the calling Process) would show the whole diagram in which the called Process resides (see Figure 13), but any data mapping will be only to that Process and not to any of the other Processes that might be in the called diagram



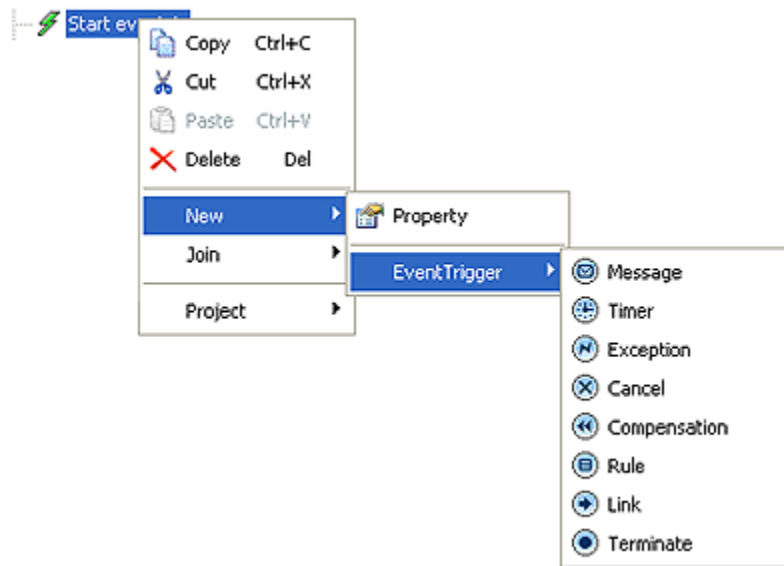
Figure 101. The whole diagram which is linked with the subProcess



## Event object actions

Start and Intermediate events have *triggers*. End events have *results*. An event can have multiple triggers (in which case it is a multiple event and it is drawn with a star symbol). To add a trigger or result to an event use the New EventTrigger/Result action.

Figure 102. Event actions



## New Property

This action created a property object in the event

## New Message Trigger

This action adds a message trigger to the event (Start, Intermediate and End events).

## New Timer Trigger

This action adds a timer trigger to the event. (Start and Intermediate events).

## New Exception Trigger

This action adds an exception trigger to the event. (Intermediate and End events).

## New Cancel Trigger

This action adds a cancel trigger to the event. (Intermediate and End events).

## New Compensation Trigger

This action adds a compensation trigger to the event. (Intermediate and End events).

## New Rule Trigger

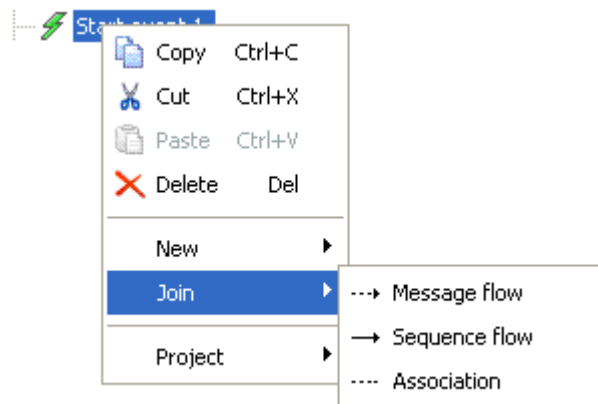
This action adds a rule trigger to the event. (Start and Intermediate events).

## New Link Trigger

This action adds a link trigger to the event. (Start, Intermediate and End events).

## New Terminate Trigger

This action adds a terminate trigger to the event. (End events).



## Join Message flow

This action joins the event to another flow object with a Message flow connection.

## Join Sequence flow

This action joins the event to another flow object with a Sequence flow connection.

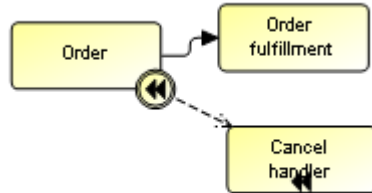
## Join Association

This action joins the event to another object with an association.

## Join Compensation Association

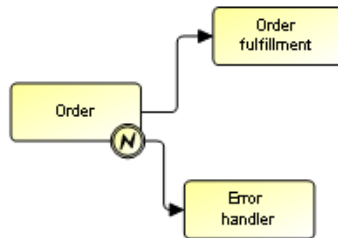
This action creates a compensation association.

The target of the compensation association displays a rewind graphical indicator to show that this activity is performed when compensation is needed for its source object (in this example “Order”).

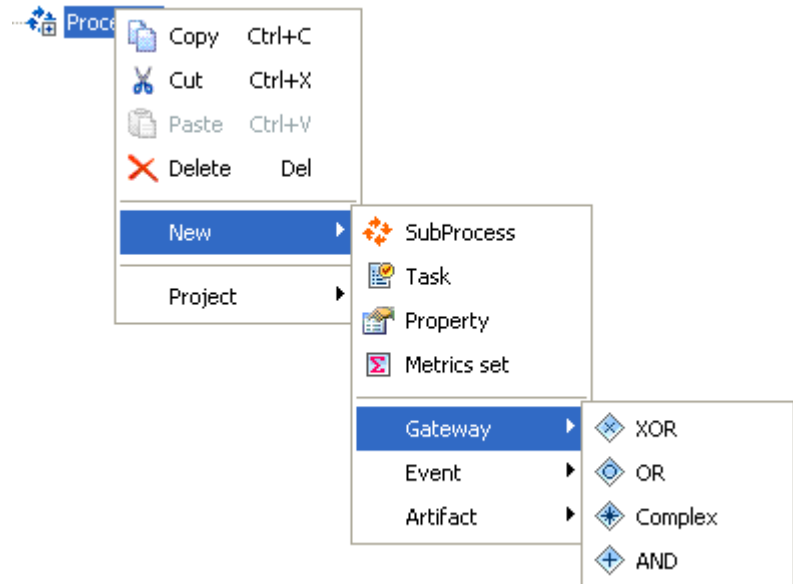


## Join Exception Flow

This action creates an exception flow.



## Process object actions



**New SubProcess**

**New Task**

**New Property**

**New MetricsSet**

**New XOR Gateway**

**New OR Gateway**

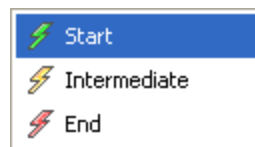
**New Complex Gateway**

**New AND Gateway**

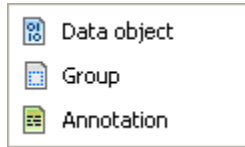
**New Start event**

**New Intermediate Event**

**New End Event**



## New Data object artifact



## New Group

## New Annotation

## Connecting objects

Connections can be made between flow objects with the BPMN Toolbar and/OR the Navigator.

### Two disconnected Task objects to be joined

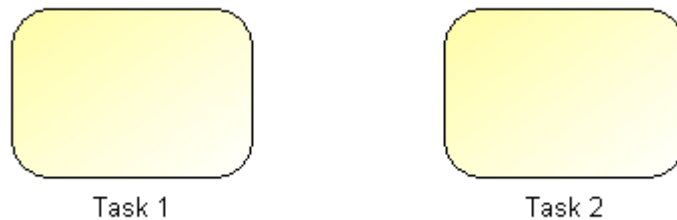
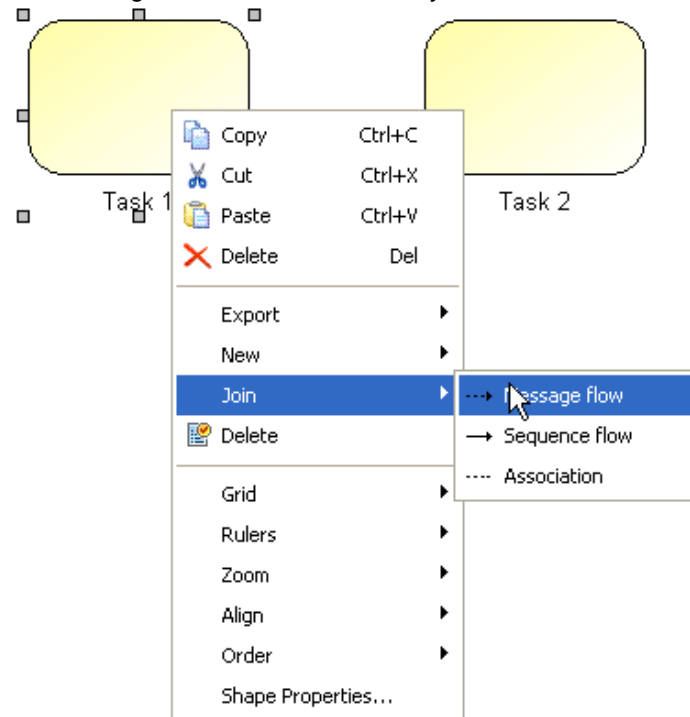


Figure 103. Right click on the source object

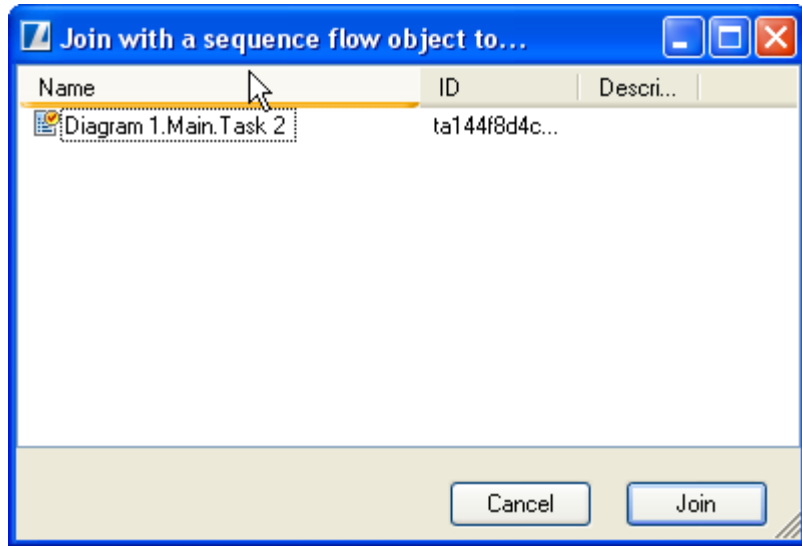




Select the appropriate type of connection. Depending upon the kind of source object you have clicked on you will see Message flow, Sequence flow and Association.

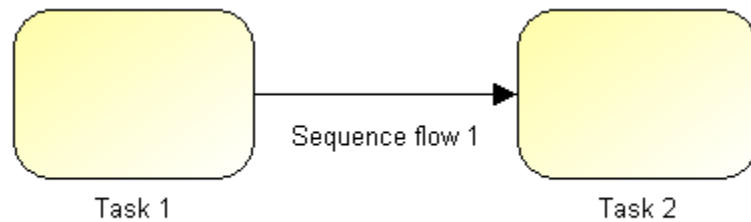
Because we want to create a Sequence flow, select the Sequence flow menu option.

**Figure 104.**The join dialog



The list of flow objects in the dialog represents the legal set of flow objects that you can join the source task with.

**Figure 105.**The Task objects are now joined by a Sequence flow



## Creating a Message flow connection between two Task objects

Figure 106. Joining tasks with message flow

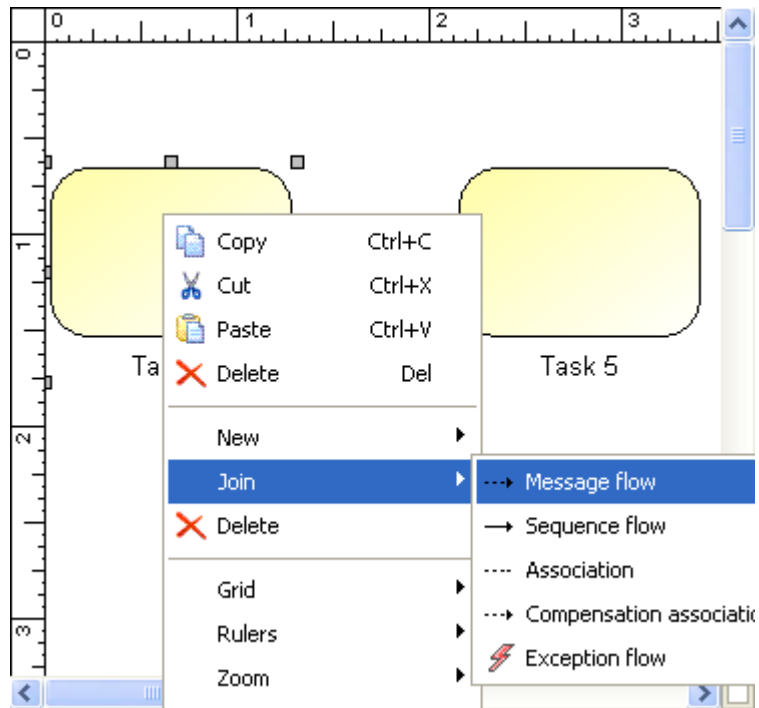
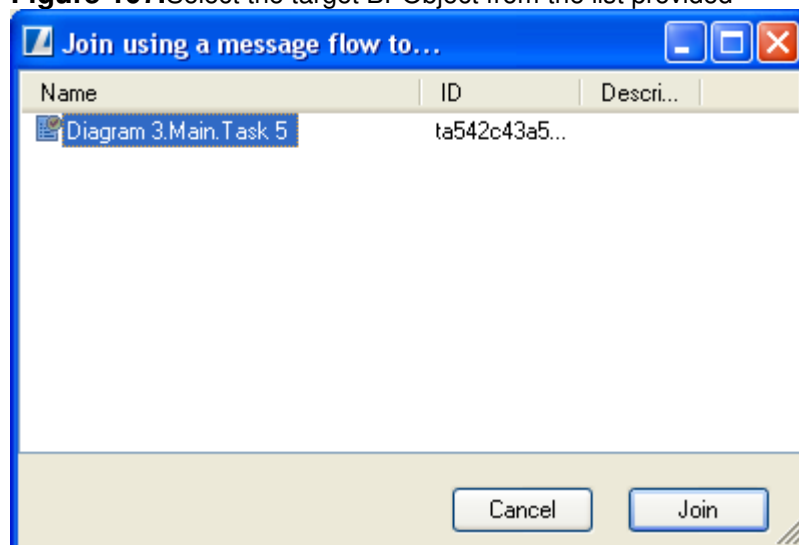


Figure 107. Select the target BPObjct from the list provided

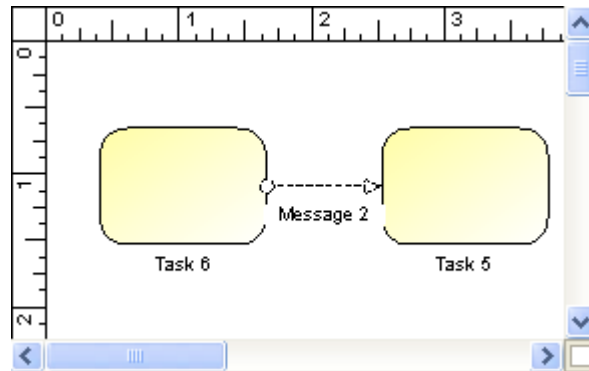




After clicking the **Join** button the two shapes will be connected with a dashed line decorated by a circular line origin and triangular arrow head. This makes it clear at a glance that it is *not* a sequence flow.

**NOTE:** You can examine the properties of the message you just created by clicking on the message link in order to select it and then switching to the *Properties* view to display its properties. .

**Figure 108.**Tasks connected with Message Flow connection



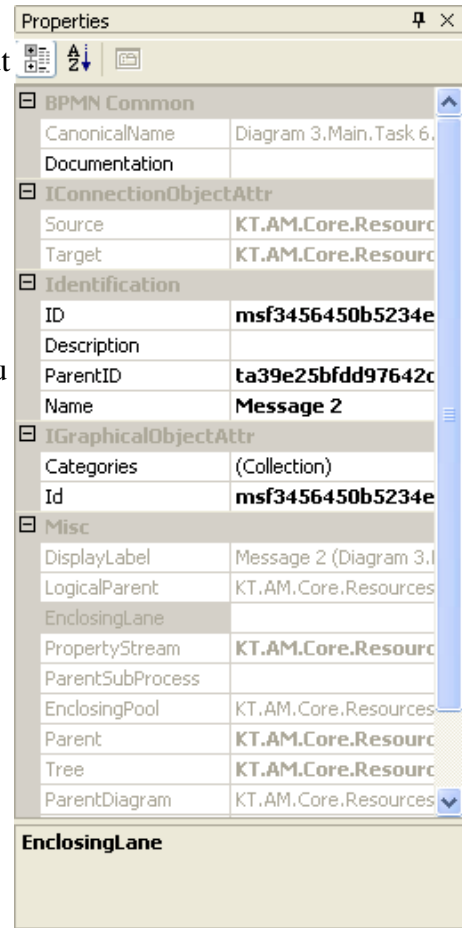
Here you can see the attributes of the message flow object.

You can change the message flow's name and documentation strings easily enough but don't change anything else.

Especially do not change an ID unless you really know and understand what you are doing because you could mess up some stuff.

If you read this warning and changed an ID anyway without understanding the ramifications then you really deserve all you get.

- Thankfully there is a Revert action context menu in .model files that can restore a broken model back to a previous unbroken generation.

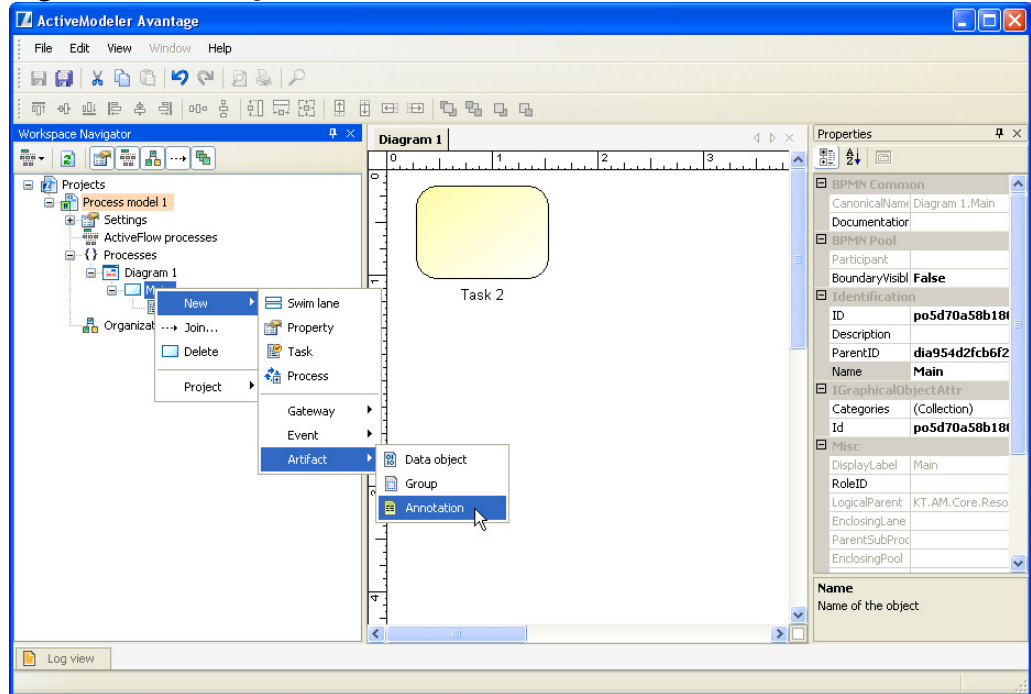


## Example: Creating an Association

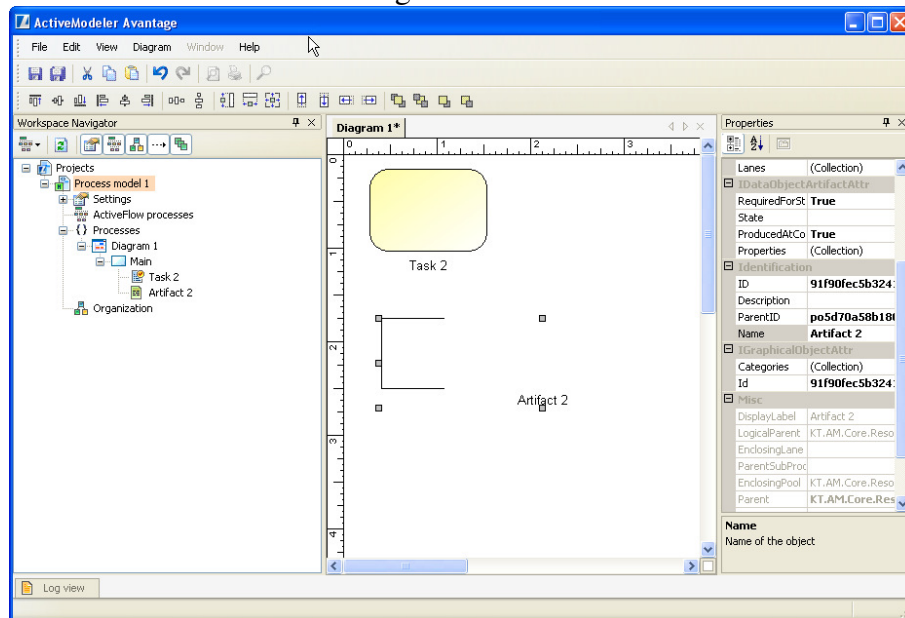
*Note: If there is no Pool or Lane visible in your diagram window then you must use the Navigator BPTree menu to insert an Annotation into the diagram. This example does that.*

This example will show you how to create a Text Annotation and associate it with a process flow object.

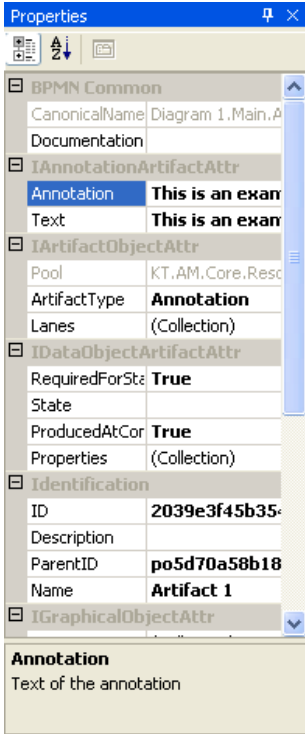
**Figure 109.**Creating an Annotation Artifact



You should now see something like this:

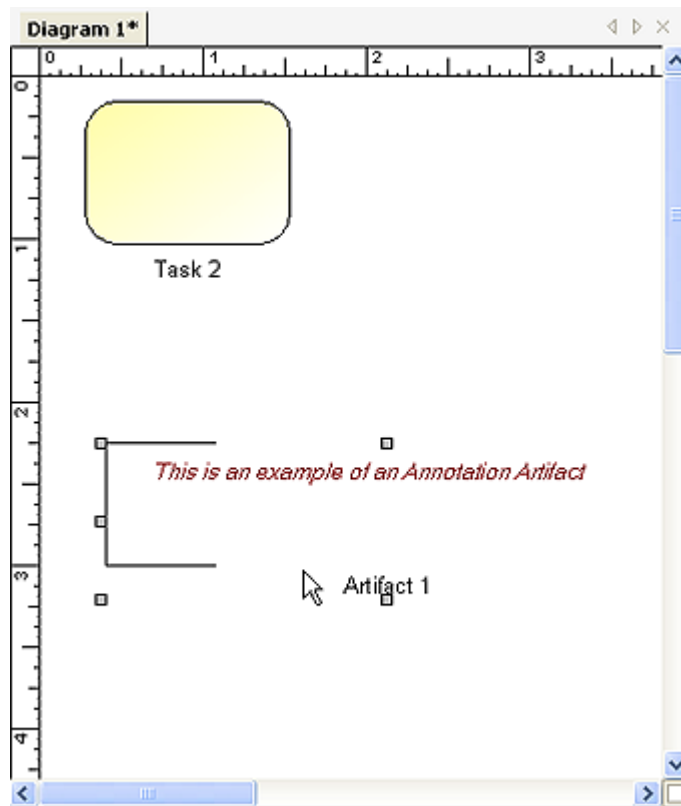


The default content of this text annotation is blank. We can easily add some text with the **Properties** inspector. Click on the annotation shape (not its label) to select the object and then click on the “*Annotation*” property in the Properties inspector.



**Figure 110.** Changing the text of an Annotation

- ➔ Enter some text into the text box of either the Annotation property or the Text property (they both reflect the same internal string)
- ➔ When you are done entering the text hit ENTER or click somewhere else in the Properties inspector.
- ➔ The diagram window will immediately display the new text of the annotation.

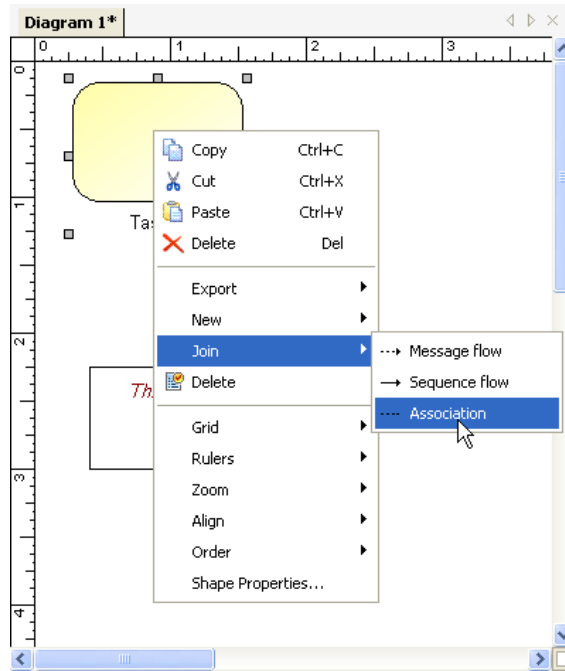




## The create association action

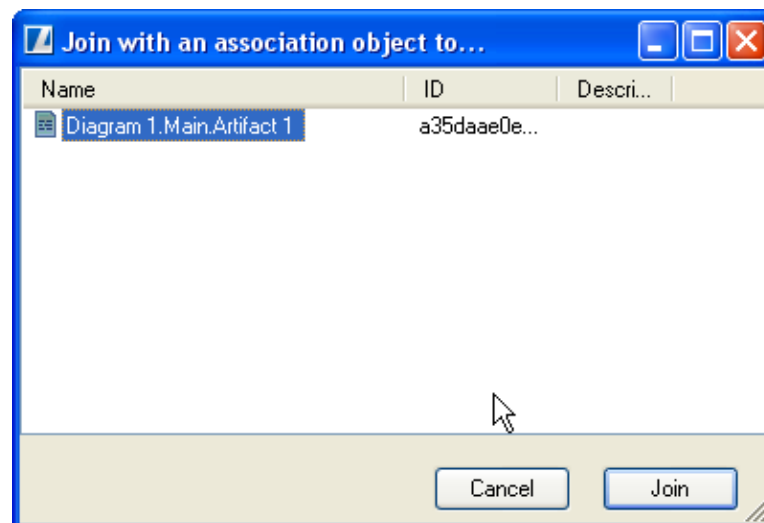
Now it is time to create the association. We will create an association between the Task object and the Annotation artifact like this.

**Figure 111.**Creating the Association



- *Right-click on the Task object*
- *Choose **Join:Association***

**Figure 112.**The join – association dialog



The dialog shows a list of the valid candidates you can join the Task to. Select the Artifact object from the list and click Join.

**Figure 113.**The completed association

# Properties

*This chapter describes how to use Property objects. Properties are objects which contain any .NET data types. Properties are key to making use of advanced functions like metrics.*

Modeler-defined Properties may be added to a Process. These Properties are “local” to the Process. All Tasks, Sub-Process objects, and Sub-Processes that are embedded shall have access to these Properties. The fully delineated name of these properties are “<process name>.<property name>” (e.g., “Add Customer.Customer Name”). If a process is embedded within another Process, then the fully delineated name SHALL also be preceded by the Parent Process name for as many Parents there are until the top level Process.

*The BPMN standard defines a property like this...*

<i>Attributes</i>	<i>Description</i>
<b>Name</b>	String Each Property has a Name (e.g., name=“Customer Name”).
<b>Type</b>	String Each Property has a Type (e.g., type=“String”).
[Type = “Set” only]	Boolean
<b>Correlation</b>	If the ConditionType attribute is set to Expression, then the ConditionExpression attribute MUST be defined. Otherwise, it is not valid. This attribute is included for mapping to BPEL4WS. The Property will map to a <i>correlationSet</i> and the child Properties will be <i>properties</i> of that <i>correlationSet</i>

## How Avantage implements properties

In addition to the BPMN standard property object, Avantage provides an extension class called a *Metricated Property*. Metricated property objects can be used for numerical analysis of a process model.

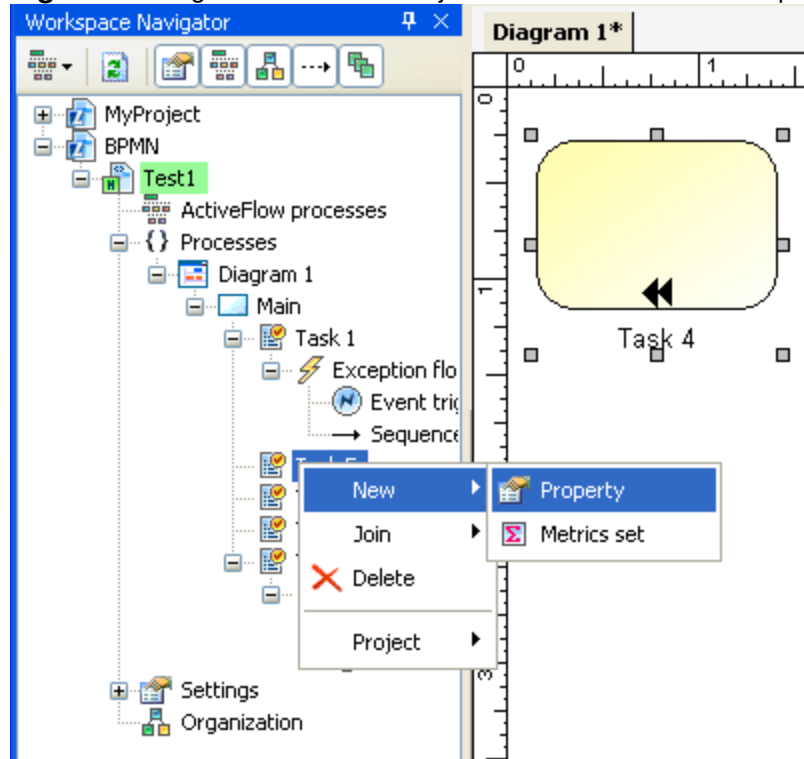
### Standard Properties

Avantage interprets the above standard expressly in terms of the .NET platform. Instead of the **Type** attribute being implemented as a **string** it is implemented as a **System.Type**.

Property names must be unique within a BP flow object.

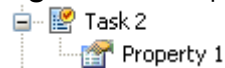
How to add a property to a flow object

**Figure 114.** Right-click on a flow object and choose the New:Property action



The new property is appears in the Navigator as a child of the flow object.

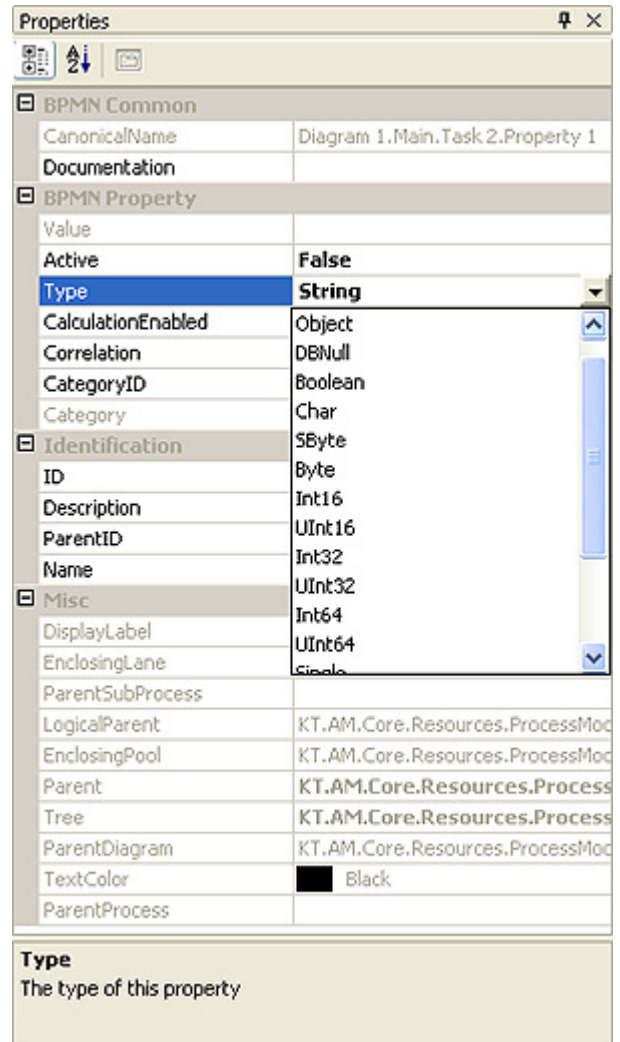
**Figure 115.** The property is displayed



1. Changing the type of a property

## Changing the type of a property

1. Select the property in the Navigator tree. This will ensure that it is displayed in the Properties Inspector view.
2. Click on the Properties view to activate it.
3. Click on the Type dropdown list
4. Choose the type from the list of types which appear in the dropdown list. The default type is *String*. If for example you wanted to store a monetary value in your property you should choose a Decimal type.



## Editing property values

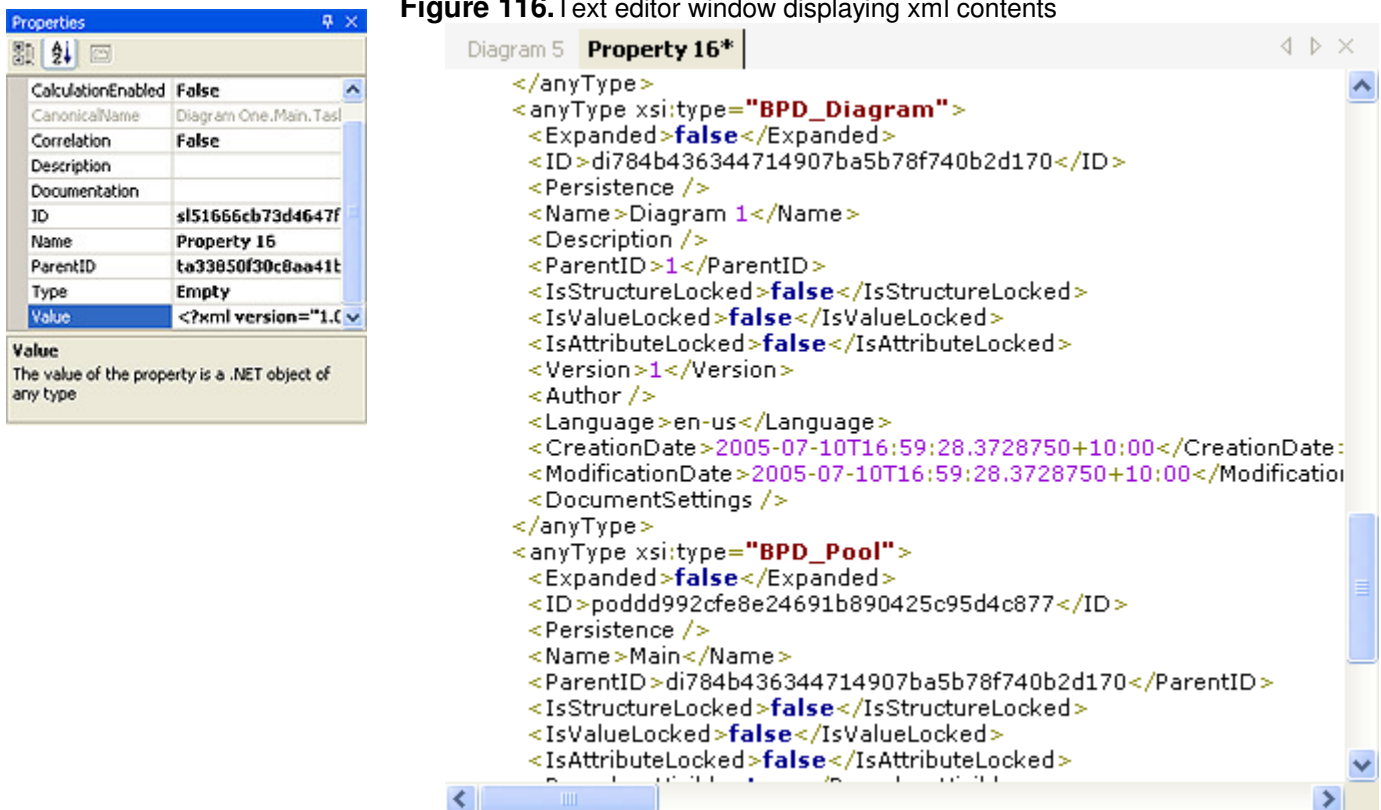
### String (text) properties.

To edit a property that contains text (i.e it is of type String) double-click on it. This will open a **text editor** window containing the contents of the property value.

The text editor treats the property as if it were a file and will prompt you to save changes back to the property object when you close it.

The text editor will automatically syntax highlight the text if it is a structured type like Xml or C#, JavaScript, ASP (Active Server Pages) and more.

**Figure 116.**Text editor window displaying xml contents



### Other properties

To change the value of any other type of property select the property so that it is visible in the *Properties* inspector and then directly change the value field.

*Note:* In order to add text to an empty String property it is necessary to use the double-click method.

## Measure points and metrics

{ }

A measure point is a position in a BPTree from which you can watch the metrics in your process model. Although Measure points are displayed in the BPTree they are transparent to the process model itself and they play no active part in the process definition.

**Figure 117.** The root of a process model is a measure point. It is called “Processes”



The Advantage **Metrics Plugin** (which must be commercially licensed from KAISHA-Tec) extends the default actions which are provided.

With the advantage metrics plugin you can select a measure point to display an reporting or analysis view for that point in the process model.



# The Log Viewer

*The Log Viewer displays the contents of the platform log file in an easy to understand format.*

## The Platform.log file

Sometimes, during the operation of Avantage or one of its plugins an error may occur. If this happens the error is recorded in the log file. This file is named `Platform.log` which is located in the `.config` directory of Avantage's installation directory.

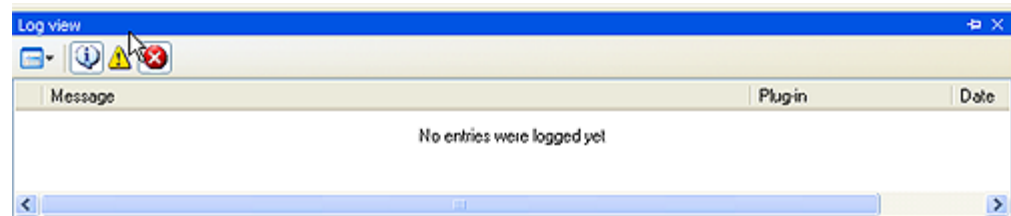
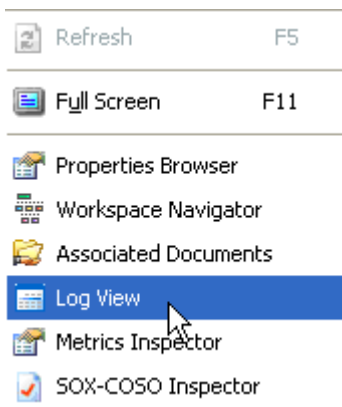
Each time you start Avantage a Session entry is made in the log file which records the date and time of the session and the command line options that were present when the application was started. Log file entries in Xml format can be read with a text file viewer but it is a lot easier to display it with the Log Viewer.

The Platform.log file is an invaluable tool in diagnosing problems with an installation or a plugin. If you are experiencing a technical problem with Avantage or a plugin you may be asked by one of our support engineers to send them the log file by email. If you are asked to do so, please ZIP the log file as this way the file is as small as possible and most importantly the contents of the file will not become corrupted during the transmission process.

## Starting the Log Viewer

To start the Log Viewer choose the **View:Log Viewer** menu action from the main menu bar.

**Figure 118.**The Log view



The log view shows the Message contents, the name of the Plugin which logged the event and the date/time when the event occurred.



## Log Viewer Filters

All of the buttons work on various aspects of filtering the information in the log file. There are three kinds of entries recorded in the log.

1. Information

This is just information that a plugin thinks is a good idea to log. It might be stuff like information describing the settings being used, noting an environmental parameter.

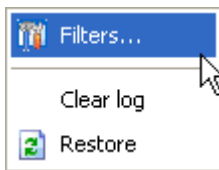
2. Warnings

These messages warn you of problems or important events that you might need to know about.

3. Errors

These messages describe severe errors that have occurred.

You can change the filtering with the toolbar buttons. Clicking on one of the three message type icons enables the display of that type of message in the log.

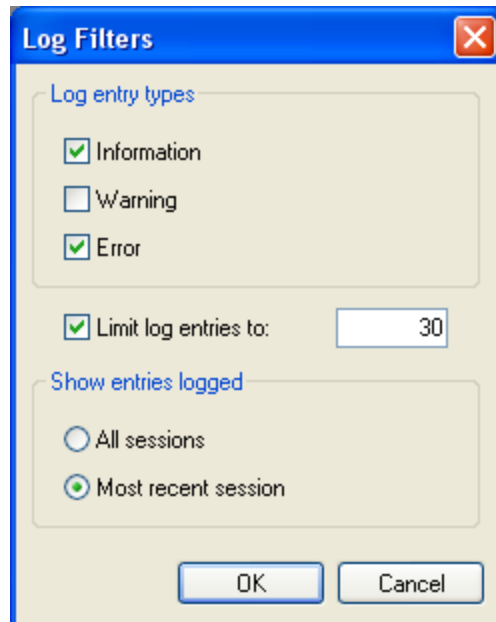


**Figure 119.**Log viewer toolbar buttons



The drop-down button allows you to clear and restore the log view and edit/modify your filter settings preferences.

**Figure 120.**Log Filters drop-down



# Organization Modeling

*The BPMN 1.0 standard omits the specification of organizational structures. Of course Pools and Swimlanes can be used to represent this but it is often not enough for real world use. Advantage provides full organization modeling which is transparently mapped onto the BPMN process model.*

## The organizational structure

The organization structure is a sub-tree within the navigator. It consists of Roles and Departments. Departments nest hierarchically. Roles are defined within departments.

The screenshot shows the 'Workspace Navigator' window with a tree view of an organization structure for 'New Piston Production'. The tree includes departments like 'Inspection', 'Factory', 'Fabrication', 'Planning', and 'Laboratory'. Each department contains roles such as 'Inspector', 'Inspector Foreman', 'Plating Operator', 'Machinist - Turner', etc. Some roles are inactive, indicated by a greyed-out icon. A red box highlights the 'Inspection' role icon with the label 'Role (human or machine)'. A red box highlights the 'Factory' department name with the label 'Department'. A red box highlights the 'Inspection' department icon with the label 'Root Department'. A red box highlights the 'Show/Hide organization information structure' icon with the label 'Show/Hide organization information structure'. A red box highlights a greyed-out role icon with the label 'Inactive organization object'. A red box highlights a scroll bar with the label 'organization file whose type is'.

*Note: The Advantage Metrics plugin makes use of organization files. This plugin stores costing lookup data for roles and departments here so that it is able to automatically determine role and rate based costs for activities which are defined with pools and lanes.*

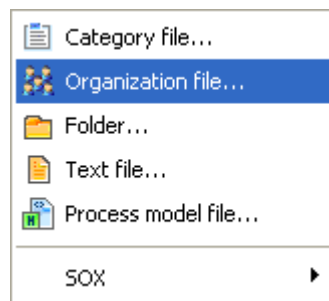
Organization files can be shared between different process models within a project and it is possible to have multiple organization files inside your project. Organization files can also be loaded and saved from an ActiveFlow database.

## Adding organization files to a project

### To add a new file

Select the desired project and right-click on its project icon, then choose the **New** action and select **Organization file...**

**Figure 121.** The New Organization file menu



You are then prompted to choose a name for the file and then the file is created in the project.

### To add an existing file

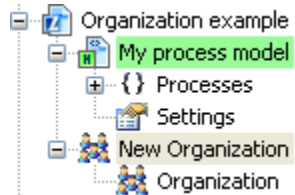
Select the desired project and right-click on its project icon, then choose the **Add** action and select **Existing organization file...**

You will be prompted to locate the file and then click OK and the file displayed in the project.

## Working with organization files

Organization files expand into a tree just like process model files. The root item of the organization is the name of the file. You can rename this item at any time.

**Figure 122.**An empty organization file



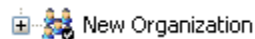
By default, the top most node in the organization structure is named “Organization”. This item represents the top level department of the organization structure and usually represents the company or organization itself. You can rename it if you wish; for example you could rename it to “My Company”.

## Enabling an organization file

After the organization file is inside your project you must enable it before it to be used by a process model, this is done by right-clicking on the file's icon and choosing the “**Set as default Organization file**” action.

When an organization file is the default within a project there is a small tick mark in the lower right hand corner of the icon.

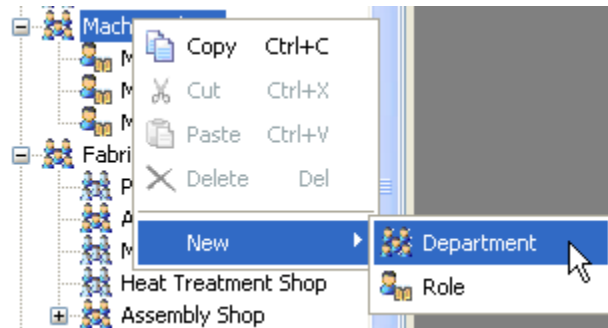
**Figure 123.**Indicating the default organization



You can have multiple organization files in your project, but only one can be used at any one time – this is the default file. This is very handy if you are developing a workflow. You might have a small scale test organization while you are developing and debugging your automated process and another full sized organization file for production. You can switch organization structures simply by setting the default file.

## Adding departments

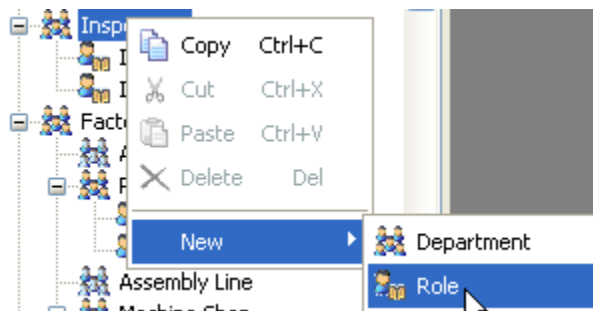
1. Select the parent department within which the department will be created.



2. Right click on the to-be parent department and choose the **New:Department** action.

## Adding roles

1. Select the parent department within which the role will be created.

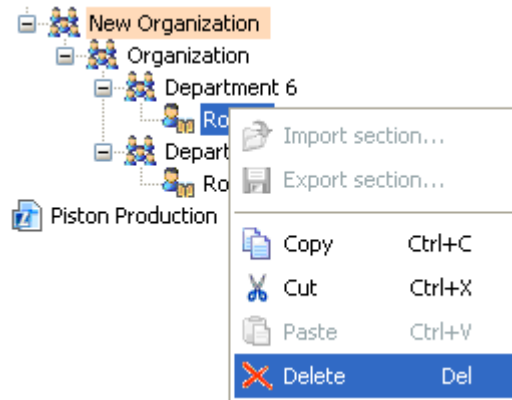


2. Right-click on the department and choose the **New:Role** action. A new role is added.

## Deleting organization units

Select the department or role to be deleted. The delete action is enabled if the object does not contain any departments or roles as children.

**Figure 124.**Select the object and hit DEL or choose Delete from the object's context menu.



The first time you 'delete' an active department or role instead of actually deleting it, Avantage makes it inactive. The result is that instead of removing the organization unit from the model it is *Ghosted* (inactivated). The reason for this is that ActiveFlow process automation engine must be able to continue working with in-flight transactions even if the organization structure has been altered.

**Figure 125.**Inactive (ghosted) organization unit.



To completely remove the object from the process model you must delete the inactivated object.



## Linking BPMN pools and swimlanes to the organization structure

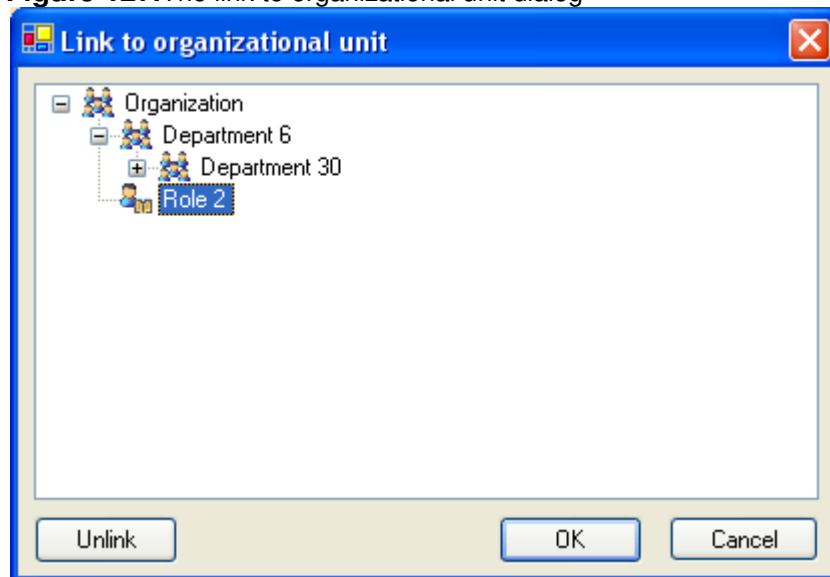
The **Join:Organizational unit** action is provided for Pool and Lane objects. You can select this action from the diagram editor (by right-clicking on the Pool or Lane) or from the Navigator (by right-clicking on the Pool or Lane icon).

The Join action creates a linkage between the BPMN object and the organization model.

**Figure 126.**Swimlane before being linked



**Figure 127.**The link to organizational unit dialog



Select the organizational unit you wish to link to the pool or lane and then click the OK button.

**Figure 128.**After linking



After linking the label of the BPMN object is decorated with the canonical path name of the organizational unit it is linked to.

## Unlinking

To unlink a BPMN object, right-click on the pool/lane in order to display the link to organization dialog, then click on the **Unlink** button.

# Working with sections

*A section is a data format used to describe a slice of a process model. Sections are used in Clipboard data and for exporting and importing process model snippets to files. Sections allow you to define BPMN structure templates that can be re-used in projects.*

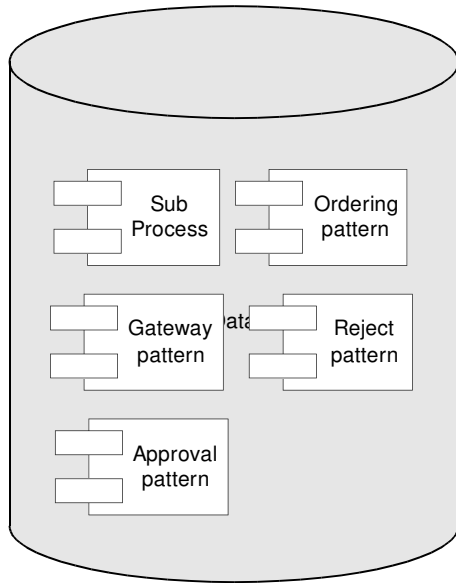
## What is a section?

A section is a collection of BPMN objects that has been cloned off from the current selection in the process model workspace. If a selected BPMN object has children then all of its children are automatically included in the section.

A section can contain a single object or a complex network of objects. This makes it useful for defining and applying patterns of BPMN shapes. You can define standard process constructs appropriate for your process modeling environment/culture.

For example you may develop a reuseable transaction handler (perhaps it is an embedded subprocess). By exporting the subprocess to a section you can easily import the subprocess pattern into other process models.

**Figure 129.**Using sections as pattern templates

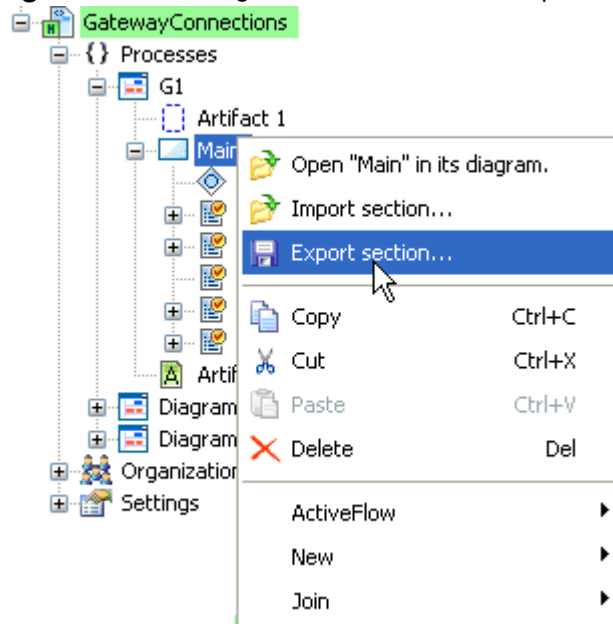


## Exporting a section

The current selection is used to anchor the position(s) in the BPTree which will be cloned into the section. If the selected object contains any children then they too will be included in the section automatically (and their children's children etc).

When you choose the Export section... action you are prompted to specify the file location particulars.

**Figure 130.**Creating a section file with the Export section action

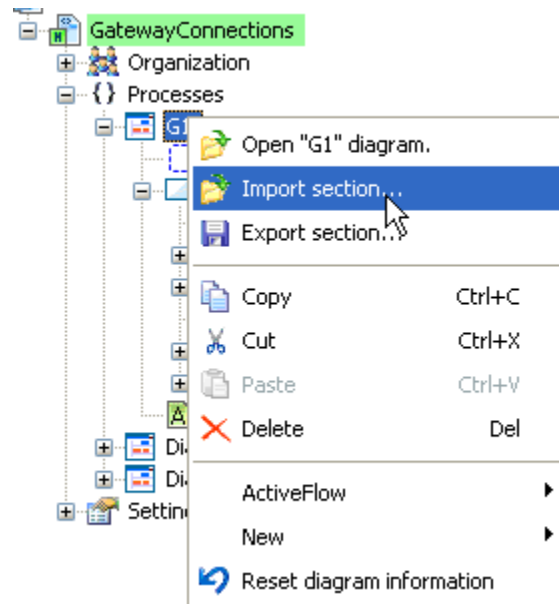


## Importing a section

This action imports a section and places it at the position in the process model that you have indicated by current selection in the workspace. The action's result is just like pasting from the clipboard but instead, the data comes from a .section file instead of the clipboard.

When you choose the Import section... action you are prompted for the location of the .section file to import.

**Figure 131.** Placing a section into a location in a process model



## Section placement rules

Certain kinds of BPMN objects may only occur as children of another specific type of BPMN object. If it is not possible to place a section at the nominated location, an error message will explain the problem.

Insuitable parent object



The object(s) in the section cannot be placed here. You can only place a section under a suitable parent. For example you cannot place an Event object underneath (as a direct child) of a diagram because event objects must occur within a pool, lane or subprocess object.

Choose another more suitable parent object for this section.

OK

# Appendix A: Process Model Settings



The settings data includes information like database connection strings, operational defaults and base metrics data.

## Main settings

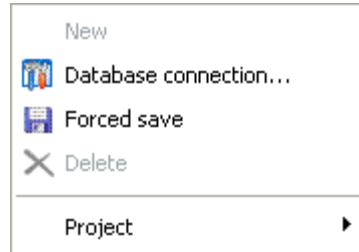
Setting	Description
<b>StorageModel</b>	This setting specifies how the process model is saved. Leave it at <i>Xml</i> unless otherwise specifically instructed.
<b>DatabaseEnabled</b>	This setting if <b>True</b> allows you to import an ActiveFlow database into your process model. After the <i>Xml</i> component of the project model file has been loaded, if this setting is <b>True</b> Avantage will load the AM/AF process model from the database in <i>ConnectionString</i> . This setting is a <i>one shot</i> operation. After a successful database read, this setting is set to <b>False</b>
<b>ConnectionString</b>	Database connection string.
<b>BackupGenerations</b>	This integer controls a rotational snapshot backup system. If, at the time a <i>.model</i> file is saved and <i>BackupGenerations</i> is <i>non zero</i> then a backup image of the <i>.model</i> file is made immediately before the file is updated. The default setting is 5, which allows the process model to keep 5 aged backup copies in the form of <i>.model</i> , <i>.model.1</i> , <i>.model.2</i> , <i>.model.3</i> , up to <i>.model.BackupGenerations</i> . This allows the process model to be reverted to one of 5 most recently saved copies in the event of file corruption, IO error or other unfortunate event



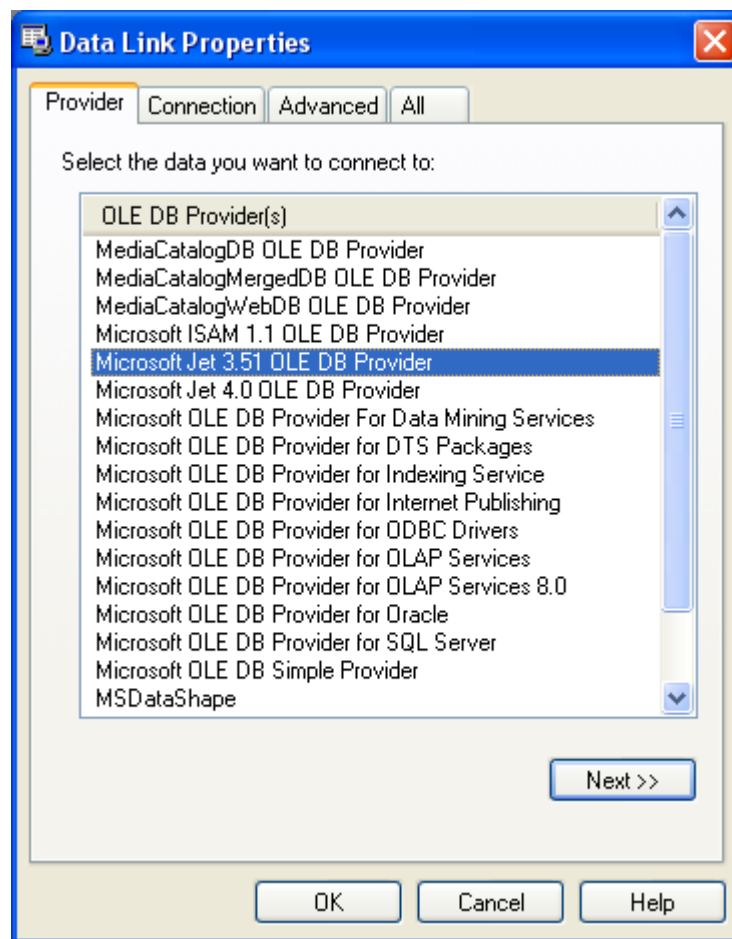


## Database connection chooser

Database connections can be chosen by right-clicking on the *Settings* icon.



**Figure 132.**Settings actions



**NOTE:** See the section on Importing from ActiveFlow for detailed information on how to do this.

## Forcing a settings change to persist immediately

You can force a save of the process model file at any time by choosing the *Settings:Forced save* action from the context menu.





## Metrics Settings

The metrics settings provide a set of base values for *Metrics Analysis*. Unless you have the *Metrics Plugin* installed these values are not used.

# Appendix B: File types

*This section describes the various types of files that Avantage uses.*

## File types in projects

Icon	File type	Description
	<i>.amproject</i> and <i>.amworkspace</i>	Project and Workspace files (Xml encoded)
	<i>.model</i>	Process model file (Xml encoded). <i>.model</i> files may can contain both ActiveFlow and BPMN process model data. Alternatively a <i>.model</i> file may contain a URL which refers to a remote process model.
	<i>.bpmn</i>	Public BPMN encoded file (Xml encoded). The demonstration of Avantage does not include support for public encoding.
	<i>.section</i>	A transactionable slice of a process model (Xml encoded). <i>.section</i> files contain an editable chunk of a <i>.model</i> . Sections are used to import and export data between <i>.model</i> files and repositories

# Appendix C: BPMN Complete set

*The complete set of BPMN entities is described here in this section.*

**Note:** In the BPMN standard, flow objects like Events, Tasks, SubProcesses, Pools and others have formally defined and named properties.

The Advantage BPMN process model implements this standard.





After creating a flow object with the **BPMN toolbar** it is sometimes necessary to use the **Property Inspector** in order to tweak a BPMN object into its final form.

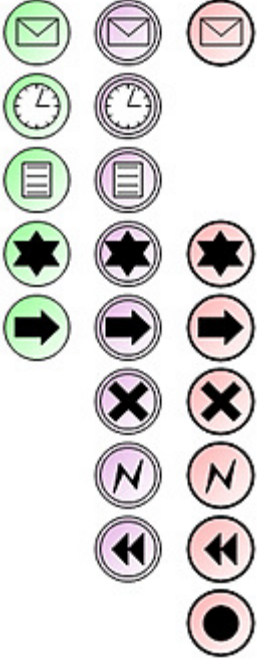
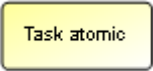
In this table, the specific property/attribute settings necessary to display a symbol notation are shown for each element variation like this...

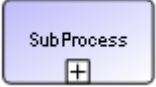
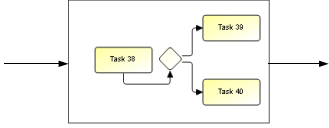

*LoopType=Standard  
MarkerVisible=True*


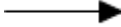
If you are unsure about how to use a particular flow object you should consult the BPMN V1.0 Specification document. It fully describes the meaning and use of the BPMN properties which are which you can see in the Property Inspector.




## BPMN Complete Set

<b>Element</b>	<b>Description</b>	<b>Notation</b>
Event	An event is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow.	
<p style="text-align: center;">Class of event</p> <p style="text-align: center;"><b>Start</b> (Message, Timer, Rule, Multiple, Link)</p> <p style="text-align: center;"><b>Intermediate</b> (Message, Timer, Rule, Multiple, Link, Cancel, Exception, Compensation)</p> <p style="text-align: center;"><b>End</b> (Message, Rule, Multiple, Link, Exception, Compensation, Terminate, Cancel)</p>	<p>As the name implies, the Start Event indicates where a particular process will start.</p> <p>Intermediate Events occur between a Start Event and an End Event. It will affect the flow of the process, but will not start or (directly) terminate the process.</p> <p>As the name implies, the End Event indicates where a process will end.</p>	  

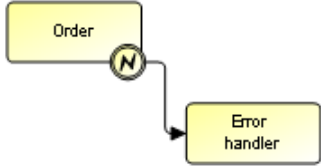
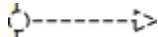
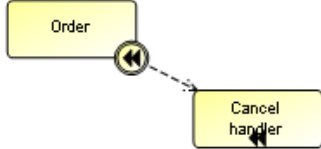

<b>Element</b>	<b>Description</b>	<b>Notation</b>
<p>Type dimension (e.g. Message, Error, Cancel etc).</p>	<p>Start and most Intermediate Events have “Triggers” that define the cause for the event. There are multiple ways that these events can be triggered. End Events may define a “Result” that is a consequence of a Sequence Flow ending.</p>	
<p>Task (Atomic)</p>	<p>A Task is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail</p>	
<p>Process/Sub-Process (non-atomic)</p>	<p>A Sub-Process is a compound activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail (a Process) through a set of subactivities.</p>	<p><i>See notations below</i></p>

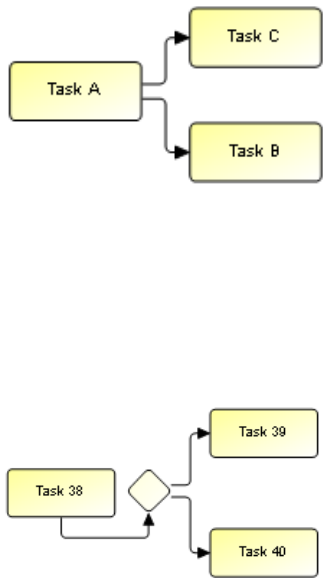
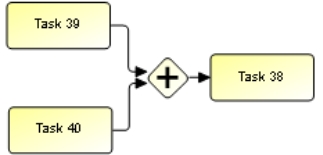
<b>Element</b>	<b>Description</b>	<b>Notation</b>
Collapsed Sub-Process	The details of the Sub-Process are not visible in the Diagram. A “plus” sign in the lower-center of the shape indicates that the activity is a Sub-Process and has a lower-level of detail.	
Expanded Sub-Process	The boundary of the Sub-Process is expanded and the details (a Process) are visible within its boundary. Note that Sequence Flow cannot cross the boundary of a Sub-Process.	
Gateway	A Gateway is used to control the divergence and convergence of multiple Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths.	

<b>Element</b>	<b>Description</b>	<b>Notation</b>
<p>Gateway Control Types</p> <p><i>MarkerVisible=True</i></p>	<p>Icons within the diamond shape will indicate the type of flow control behavior. The types of control include:</p> <ul style="list-style-type: none"> <li>• XOR -- exclusive decision and merging. Both Data-Based and Event-Based. Data-Based can be shown with or without the "X" marker.</li> <li>• OR -- inclusive decision and merging</li> <li>• Complex -- complex conditions and situations (e.g., 3 out of 5)</li> <li>• AND -- forking and joining. Each type of control affects both the incoming and outgoing Flow.</li> </ul>	
<p>Sequence Flow</p>	<p>A Sequence Flow is used to show the order that activities will be performed in a Process.</p>	
<p>Normal Flow</p>	<p>Normal Sequence Flow refers to the flow that originates from a Start Event and continues through activities via alternative and parallel paths until it ends at an End Event.</p>	

<b>Element</b>	<b>Description</b>	<b>Notation</b>
Uncontrolled flow	Uncontrolled flow refers to flow that is not affected by any conditions or does not pass through a Gateway. The simplest example of this is a single Sequence Flow connecting two activities. This can also apply to multiple Sequence Flow that converge on or diverge from an activity. For each uncontrolled Sequence Flow a "Token" will flow from the source object to the target object.	
Conditional flow <i>ConditionType=Expression</i>	Sequence Flow can have condition expressions that are evaluated at runtime to determine whether or not the flow will be used. If the conditional flow is outgoing from an activity, then the Sequence Flow will have a minidiamond at the beginning of the line (see figure to the right). If the conditional flow is outgoing from a Gateway, then the line will not have a minidiamond (see figure in the row above).	
Default flow <i>ConditionType=Default</i>	For Data-Based Exclusive Decisions or Inclusive Decisions, one type of flow is the Default condition flow. This flow will be used only if all the other outgoing conditional flow is not true at runtime. These Sequence Flow will have a diagonal slash will be added to the beginning of the line (see the figure to the right).	

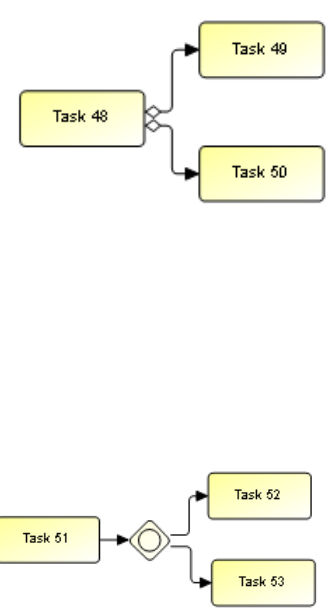


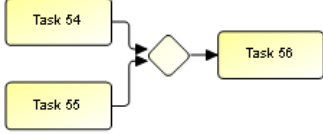
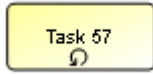
<i>Element</i>	<i>Description</i>	<i>Notation</i>
Exception Flow	Exception Flow occurs outside the Normal Flow of the Process and is based upon an Intermediate Event that occurs during the performance of the Process.	
Message Flow	A Message Flow is used to show the flow of messages between two entities that are prepared to send and receive them. In BPMN, two separate Pools in the Diagram will represent the two entities.	
Compensation Association	Compensation Association occurs outside the Normal Flow of the Process and is based upon an event (a Cancel Intermediate Event) that is triggered through the failure of a Transaction or a Compensate Event. The target of the Association must be marked as a Compensation Activity.	
Data Object	Data Objects are considered Artifacts because they do not have any direct effect on the Sequence Flow or Message Flow of the Process, but they do provide information about what activities require to be performed and/or what they produce.	

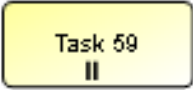
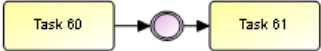

Element	Description	Notation
<p>Fork (AND-Split)</p>	<p>BPMN uses the term “fork” to refer to the dividing of a path into two or more parallel paths (also known as an ANDSplit). It is a place in the Process where activities can be performed concurrently, rather than sequentially. There are two options:</p> <p>Multiple Outgoing Sequence Flow can be used (see figure top-right). This represents “uncontrolled” flow is the preferred method for most situations.</p> <p>A Parallel (AND) Gateway can be used (see figure bottom-right). This will be used rarely, usually in combination with other Gateways.</p>	
<p>Join (AND-Join)</p>	<p>BPMN uses the term “join” to refer to the combining of two or more parallel paths into one path (also known as an AND-Join or synchronization). A Parallel (AND) Gateway is used to show the joining of multiple Flow.</p>	
<p>Decision, Branching Point (OR-Split)</p>	<p>Decisions are Gateways within a business process where the flow of control can take one or more alternative paths.</p>	<p><i>(See next five rows)</i></p>

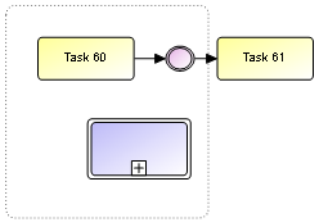

<b>Element</b>	<b>Description</b>	<b>Notation</b>
<p>Exclusive</p> <p><i>XORType=Data</i></p> <p><i>XORType=Event</i></p>	<p>An Exclusive Gateway (XOR) restricts the flow such that only one of a set of alternatives may be chosen during runtime. There are two types of Exclusive Gateways:</p> <p>Data-based and Event-based.</p>	<p>The notation shows two types of Exclusive Gateways. The top type is labeled 'Data based' and is represented by a yellow diamond. The bottom type is labeled 'Event based' and is represented by a diamond with a star in the center.</p>
<p>Data-based</p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow. Only one of the Alternatives will be chosen.</p>	<p>The diagram shows a yellow rounded rectangle labeled 'Task 41' on the left. An arrow points from Task 41 to a yellow diamond-shaped Exclusive Gateway. From the gateway, two arrows branch out to the right. The top arrow is labeled 'Default' and points to a yellow rounded rectangle labeled 'Task 42'. The bottom arrow points to a yellow rounded rectangle labeled 'Task 43'.</p>

<b>Element</b>	<b>Description</b>	<b>Notation</b>
<p>Event-based</p> <p><i>XORType=Event</i></p>	<p>This Decision represents a branching point where Alternatives are based on an Event that occurs at that point in the Process. The specific Event, usually the receipt of a Message, determines which of the paths will be taken. Other types of Events can be used, such as Timer. Only one of the Alternatives will be chosen.</p> <p>There are two options for receiving Messages:</p> <p>Tasks of Type Receive can be used (see figure top-right).</p> <p>Intermediate Events of Type Message can be used (see figure bottom-right).</p>	

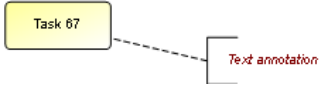
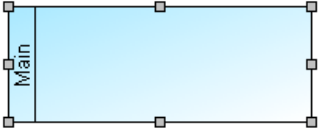
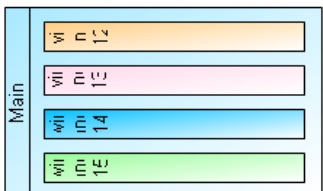
Element	Description	Notation
<p>Inclusive</p> <p><i>ConditionType=Expression</i></p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow.</p> <p>In some sense it is a grouping of related independent Binary (Yes/No) Decisions. Since each path is independent, all combinations of the paths may be taken, from zero to all.</p> <p>However, it should be designed so that at least one path is taken. A Default Condition could be used to ensure that at least one path is taken.</p> <p>There are two versions of this type of</p> <p>Decision: The first uses a collection of conditional Sequence Flow, marked with mini-diamonds (see top-right figure).</p> <p>The second uses an OR Gateway, usually in combination with other Gateways (see bottom-right picture).</p>	

Element	Description	Notation
<p>Merging (OR-Join)</p> <p><i>MarkerVisible=False</i></p>	<p>BPMN uses the term “merge” to refer to the exclusive combining of two or more paths into one path (also known as an a OR-Join).</p> <p>A Merging (XOR) Gateway is used to show the merging of multiple Flow.</p> <p>If all the incoming flow is alternative, then a Gateway is not needed. That is, uncontrolled flow provides the same behavior.</p>	
<p>Looping</p>	<p>BPMN provides 2 (two) mechanisms for looping within a Process.</p>	<p><i>(See next two figures)</i></p>
<p>Activity Looping</p> <p><i>LoopType=Standard</i></p>	<p>The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. There are two types of loops:</p> <p>Standard and Multi-Instance. A small looping indicator will be displayed at the bottom-center of the activity.</p>	
<p>Sequence Flow Looping</p>	<p>Loops can be created by connecting a Sequence Flow to an “upstream” object. An object is considered to be upstream if that object has an outgoing Sequence Flow that leads to a series of other Sequence Flow, the last of which is an incoming Sequence Flow for the original object.</p>	

<b>Element</b>	<b>Description</b>	<b>Notation</b>
<p>Multiple Instances</p> <p><i>LoopType=MultiInstance</i></p>	<p>The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. A small parallel indicator will be displayed at the bottom-center of the activity.</p>	
<p>Process Break (something out of the control of the process makes the process pause)</p>	<p>A Process Break is a location in the Process that shows where an expected delay will occur within a Process. An Intermediate Event is used to show the actual behavior (see top-right figure). In addition, a Process Break Artifact, as designed by a modeler or modeling tool, can be associated with the Event to highlight the location of the delay within the flow.</p>	
<p>Transaction</p> <p><i>IsATransaction=True</i></p>	<p>A transaction is a Sub-Process that is supported by special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled.</p> <p>The attributes of the activity will determine if the activity is a transaction. A double-lined boundary indicates that the Sub-Process is a Transaction.</p>	

Element	Description	Notation
Nested/Embedded Sub-Process (Inline Block)	A nested (or embedded) Sub-Process is an activity that shares the same set of data as its parent process. This is opposed to a Sub-Process that is independent, re-usable, and referenced from the parent process. Data needs to be passed to the referenced Sub-Process, but not to the nested Sub-Process.	There is no special indicator for nested Sub-Processes
Group (a box around a group of objects for documentation purposes)	A grouping of activities that does not affect the Sequence Flow. The grouping can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools.	
Off-Page Connector	Generally used for printing, this object will show where the Sequence Flow leaves one page and then restarts on the next page. A Link Intermediate Event can be used as an Off-Page Connector.	(Not implemented in Avantage Beta 1)
Association <i>Direction={None, From, To, Both}</i>	An Association is used to associate information with Flow Objects. Text and graphical non-Flow Objects can be associated with the Flow Objects.	



<b>Element</b>	<b>Description</b>	<b>Notation</b>
Text Annotation (attached with an Association)	Text Annotations are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.	
Pool  <i>BoundaryVisible={True, False}</i>	A Pool represents a Participant in a Process. It is also acts as a “swimlane” and a graphical container for partitioning a set of activities from other Pools, usually in the context of B2B situations.	
Lanes	A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities within a Pool.	

<i>Element</i>	<i>Description</i>	<i>Notation</i>

# Index

## Alphabetical Index

Actions.....	48
Activity tools.....	149
Adding departments.....	236
Adding roles.....	236
Alignment tool.....	165
Annotation.....	244
Artifact tools.....	155
Association.....	244
Associations.....	92
Auto layout.....	108
BoundaryVisible.....	77
BPMN Complete Set.....	244
BPMN toolbar.....	149
BPTree.....	74
Branching Point (OR-Split).....	244
Canvas.....	99
Canvas menus.....	171
Clipboard.....	89
connecting objects.....	92
Contributions.....	71
create a new process model.....	67

create a process diagram.....	74
Create a Project.....	63
Data Object.....	244
Default diagram editor properties.....	177
Deleting objects.....	
Organization files.....	236
Diagram Color.....	173
Diagram editor.....	98
Diagram Editor.....	
Preferences Settings.....	178
Diagram Menu.....	100
dirty.....	83
Displaying in diagram.....	
Text Annotations.....	
Document Properties.....	155
Docking.....	52
Editors.....	48
Embedded Sub-Process (Inline Block).....	244
Event.....	244
Event tools.....	151
EventTrigger.....	79
Exporting a section.....	240
Features and benefits.....	20
Files.....	47

Floating views.....	57
forcibly save.....	85
Fork (AND-Split).....	244
Gateway.....	244
Data-based.....	244
Event-based.....	244
Exclusive.....	244
Inclusive.....	244
Merging (OR-Join).....	244
Gateway tools.....	150
Group.....	244
Importing a section.....	240
Installation.....	22
structure.....	23
Join (AND-Join).....	244
Labels.....	106
Lanes.....	244
Legal Notice.....	4
Linking tools.....	160
Looping.....	244
Activity.....	244
Sequence Flow.....	244
Measurements and Size.....	174
modifications.....	88

Multiple Instances.....	244
Navigating.....	113
Navigator.....	70
Nudge tool.....	166
Object actions.....	71
Organization structure.....	231
organizational structure.....	231
Overview window.....	113
Placing flow objects.....	77
Plugins.....	48
Pool.....	244
Process Break.....	244
Processes Tree.....	69
Projects.....	47
Properties.....	218
QuickStart Guide.....	50
Revert.....	86
Rulers.....	100
Saving and coordinating changes.....	83
Section.....	240
Section placement rules.....	240
Selections.....	100
anchor.....	101
Shape actions.....	106

Sequence Flow.....244

Shape Properties.....81

Show Labels.....101

Sizing toolbar.....167

Space equalization tool.....168

Sub-Process.....244

Swimlane tools.....154

Symbol notation.....244

Symbols.....72

    repertoire.....72

Tabbed layout.....58

Task.....244

Toolbar Buttons.....70

Transaction.....244

Views.....47

Workspaces.....46

Zoom.....172

.....57, 107, 155, 159, 235

.amproject file.....66