

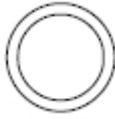

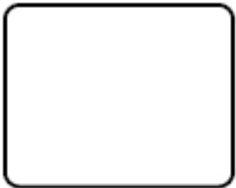
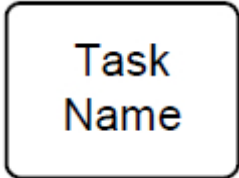
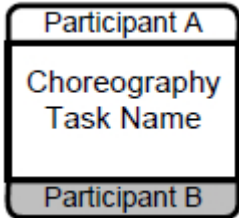
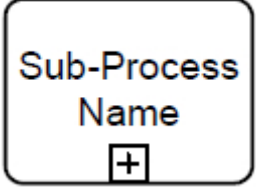
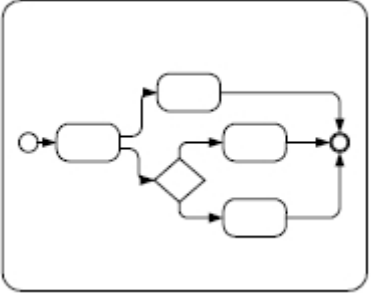
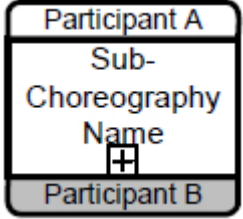
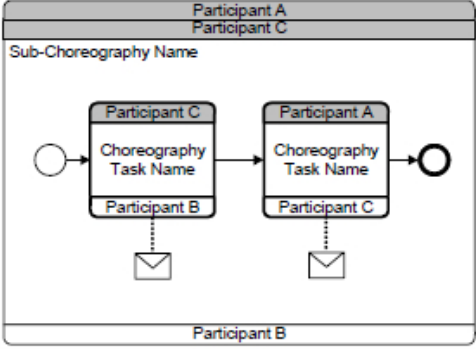



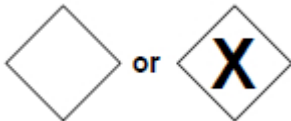





# BPMN 2.0 Notation Guide





(derived from section 7.2.2 of the BPMN Specification v2.0 available at: [www.omg.org/spec/BPMN/2.0](http://www.omg.org/spec/BPMN/2.0) )

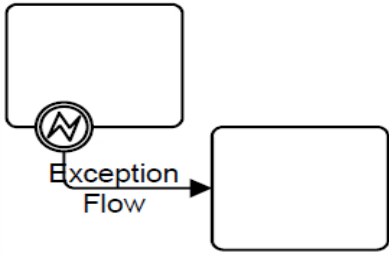

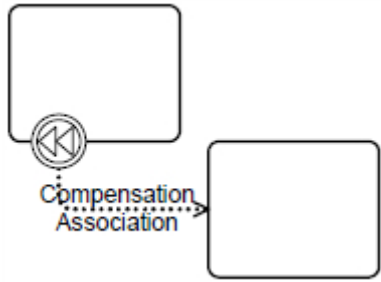





Element	Discription	Notation
Event	<p>An Event is something that “happens” during the course of a Process or a Choreography. These Events affect the flow of the model and usually have a cause (Trigger) or an impact (Result). Events are circles with open centers to allow internal markers to differentiate different Triggers or Results. There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.</p>	
<p>Flow Dimension (e.g., Start, Intermediate, End)</p> <p>Start</p> <p>Intermediate</p> <p>End</p>	<p>As the name implies, the Start Event indicates where a particular Process or Choreography will start.</p> <p>Intermediate Events occur between a Start Event and an End Event. They will affect the flow of the Process or Choreography, but will not start or (directly) terminate the Process.</p> <p>As the name implies, the End Event indicates where a Process or Choreography will en</p>	<p>Start</p>  <p>Intermediate</p>  <p>End</p> 

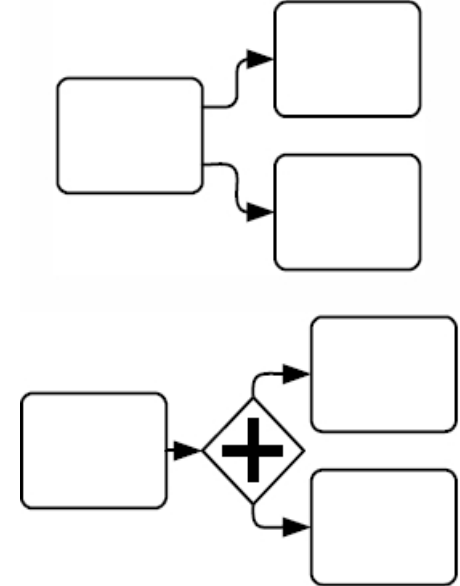
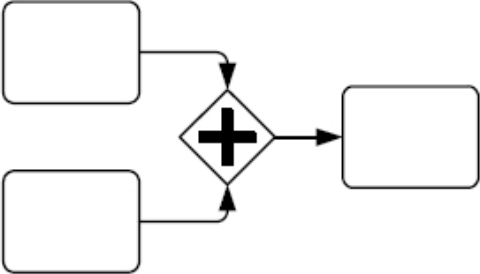
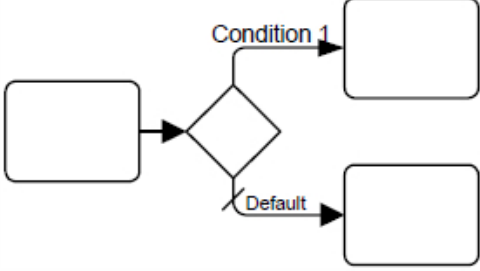
Element	Discription	Notation																																																				
<p>Type Dimension (e.g., None, Message, Timer, Error, Cancel, Compensation, Conditional, Link, Signal, Multiple, Terminate.)</p>	<p>The Start and some Intermediate Events have “triggers” that define the cause for the Event. There are multiple ways that these events can be triggered. End Events MAY define a “result” that is a consequence of a Sequence Flow path ending. Start Events can only react to (“catch”) a trigger. End Events can only create (“throw”) a result. Intermediate Events can catch or throw triggers. For the Events, triggers that catch, the markers are unfilled, and for triggers and results that throw, the markers are filled. Additionally, some Events, which were used to interrupt Activities in BPMN 1.1, can now be used in a mode that does not interrupt. The boundary of these Events is dashed.</p>	<table border="0"> <thead> <tr> <th></th> <th>“Catching”</th> <th>“Throwing”</th> <th>Non-Interrupting</th> </tr> </thead> <tbody> <tr> <td>Message</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Timer</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Error</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Escalation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Cancel</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Compensation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Conditional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Link</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Signal</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Terminate</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Multiple</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Parallel Multiple</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		“Catching”	“Throwing”	Non-Interrupting	Message				Timer				Error				Escalation				Cancel				Compensation				Conditional				Link				Signal				Terminate				Multiple				Parallel Multiple			
	“Catching”	“Throwing”	Non-Interrupting																																																			
Message																																																						
Timer																																																						
Error																																																						
Escalation																																																						
Cancel																																																						
Compensation																																																						
Conditional																																																						
Link																																																						
Signal																																																						
Terminate																																																						
Multiple																																																						
Parallel Multiple																																																						
<p>Activity</p>	<p>An Activity is a generic term for work that company performs in a Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process Model are: Sub- Process and Task, which are rounded rectangles. Activities are used in both standard Processes and in Choreographies.</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 detail.</p>																																																					
<p>Choreography Task</p>	<p>A Choreography Task is an atomic Activity in a Choreography. It represents a set of one (1) or more Message exchanges. Each Choreography Task involves two (2) <i>Participants</i>. The name of the Choreography Task and each of the <i>Participants</i> are all displayed in the different bands that make up the shape’s graphical notation. There are two (2) or more <i>Participant</i> Bands and one Task Name Band.</p>																																																					

Element	Discription	Notation
Process/Sub-Process (non-atomic)	A Sub-Process is a compound Activity that is included within a Process or Choreography. It is compound in that it can be broken down into a finer level of detail (a Process or Choreography) through a set of sub-Activities.	See Next Four Figures
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 lowerlevel 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 Flows cannot cross the boundary of a Sub-Process.	
Collapsed Sub-Choreography	The details of the Sub-Choreography are not visible in the Diagram. A “plus” sign in the lower-center of the Task Name Band of the shape indicates that the Activity is a Sub-Process and has a lowerlevel of detail.	
Expanded Sub-Choreography	The boundary of the Sub-Choreography is expanded and the details (a Choreography) are visible within its boundary. Note that Sequence Flows cannot cross the boundary of a Sub-Choreography.	

Element	Discription	Notation
Gateway	<p>A Gateway is used to control the divergence and convergence of Sequence Flows in a Process and in a Choreography. Thus, it will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behavior control (see below).</p>	
Gateway Control Types	<p>Icons within the diamond shape of the Gateway will indicate the type of flow control behavior. The types of control include:</p> <ul style="list-style-type: none"> <li>• Exclusive decision and merging. Both Exclusive and Event- Based perform exclusive decisions and merging Exclusive can be shown with or without the “X” marker.</li> <li>• Event-Based and Parallel Event-based gateways can start a new instance of the Process.</li> <li>• Inclusive Gateway decision and merging. • Complex Gateway -- complex conditions and situations (e.g., 3 out of 5).</li> <li>• Parallel Gateway forking and joining.</li> </ul> <p>Each type of control affects both the incoming and outgoing flow.</p>	<p>Exclusive</p>  <p>Event-Based</p>  <p>Parallel Event-Based</p>  <p>Inclusive</p>  <p>Complex</p>  <p>Parallel</p> 

Element	Discription	Notation
Sequence Flow	A Sequence Flow is used to show the order that Activities will be performed in a Process and in a Choreography.	See next seven figures
Normal Flow	<i>Normal flow</i> refers to paths of Sequence Flow that do not start from an Intermediate Event attached to the boundary of an Activity.	
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 Flows that converge to or diverge from an Activity. For each uncontrolled Sequence Flows a token will flow from the source object through the Sequence Flows to the target object.	
Conditional flow	A Sequence Flow can have a condition Expression that are evaluated at runtime to determine whether or not the Sequence Flow will be used (i.e., will a token travel down the Sequence Flow). If the conditional flow is outgoing from an Activity, then the Sequence Flow will have a mini-diamond at the beginning of the connector (see figure to the right). If the conditional flow is outgoing from a Gateway, then the line will not have a mini-diamond (see figure in the row above).	
Default flow	For Data-Based Exclusive Gateways or Inclusive Gateways, 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 Flows will have a diagonal slash will be added to the beginning of the connector (see the figure to the right).	

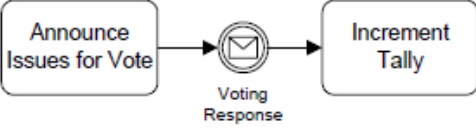



Element	Description	Notation
Exception flow	Exception flow occurs outside the normal flow of the Process and is based upon an Intermediate Event attached to the boundary of an Activity that occurs during the performance of the Process	
Message flow	A Message Flow is used to show the flow of Messages between two Participants that are prepared to send and receive them. In BPMN, two separate Pools in a Collaboration Diagram will represent the two Participants (e.g., PartnerEntities and/or PartnerRoles).	
Compensation Association	Compensation Association occurs outside the normal flow of the Process and is based upon a Compensation Intermediate Event that is triggered through the failure of a transaction or a throw Compensation Event. The target of the Association MUST be marked as a Compensation Activity.	
Data Object	Data Objects provide information about what Activities require to be performed and/or what they produce, Data Objects can represent a singular object or a collection of objects. Data Input and Data Output provide the same information for Processes.	<p data-bbox="1150 1160 1310 1193">Data Object</p>  <p data-bbox="1078 1339 1382 1373">Data Object Collection</p>  <p data-bbox="1062 1518 1398 1552">Data Input    Data Output</p>  
Message	A Message is used to depict the contents of a communication between two Participants (as defined by a business PartnerRole or a business PartnerEntity).	

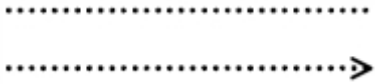
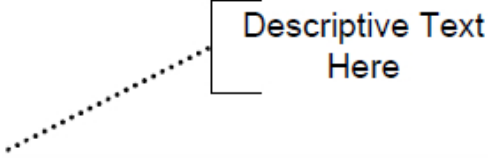


Element	Discription	Notation
Fork	<p>BPMN uses the term “fork” to refer to the dividing of a path into two or more parallel paths (also known as an AND-Split). It is a place in the Process where activities can be performed concurrently, rather than sequentially.</p> <p>There are two options:</p> <ul style="list-style-type: none"> <li>• Multiple Outgoing Sequence Flows can be used (see figure top-right). This represents “uncontrolled” flow is the preferred method for most situations.</li> <li>• A Parallel Gateway can be used (see figure bottom-right). This will be used rarely, usually in combination with other Gateways.</li> </ul>	
Join	<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 Gateway is used to show the joining of multiple Sequence Flows.</p>	
Decision, Branching Point	<p>Decisions are Gateways within a Process or a Choreography where the flow of control can take one or more alternative paths.</p>	<p>See next five rows.</p>
Exclusive	<p>This Decision represents a branching point where Alternatives are based on conditional Expressions contained within the outgoing Sequence Flows. Only one of the Alternatives will be chosen.</p>	

Element	Discription	Notation
Event-Based	<p>This Decision represents a branching point where Alternatives are based on an Event that occurs at that point in the Process or Choreography. 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> <ul style="list-style-type: none"> <li>• Tasks of Type Receive can be used (see figure top-right).</li> <li>• Intermediate Events of Type Message can be used (see figure bottom-right).</li> </ul>	
Inclusive	<p>This Decision represents a branching point where Alternatives are based on conditional Expressions contained within the outgoing Sequence Flows. 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. However, it should be designed so that at least one path is taken.</p> <p>A Default Condition could be used to ensure that at least one path is taken. There are two versions of this type of Decision:</p> <ul style="list-style-type: none"> <li>• The first uses a collection of conditional Sequence Flows, marked with mini-diamonds (see top-right figure).</li> <li>• The second uses an Inclusive Gateway (see bottom-right picture).</li> </ul>	



Element	Discription	Notation
Merging	BPMN uses the term “merge” to refer to the exclusive combining of two or more paths into one path (also known as an OR-Join). A Merging Exclusive Gateway is used to show the merging of multiple Sequence Flows (see upper figure to the right). If all the incoming flow is alternative, then a Gateway is not needed. That is, uncontrolled flow provides the same behavior (see lower figure to the right).	
Looping	BPMN provides two mechanisms for looping within a Process.	See Next Two Figures
Activity Looping	The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. There are two types of loops: Standard and Multi- Instance. A small looping indicator will be displayed at the bottom-center of the activity.	
Sequence Flow Looping	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 Flows, the last of which is an incoming Sequence Flow for the original object.	
Multiple Instances	The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. A set of three horizontal lines will be displayed at the bottom-center of the activity for sequential Multi-Instances (see upper figure to the right). A set of three vertical lines will be displayed at the bottom-center of the activity for sequential Multi-Instances (see lower figure to the right).	

Element	Discription	Notation
<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>	 <pre> graph LR     A[Announce Issues for Vote] --&gt; B((Voting Response))     B --&gt; C[Increment Tally] </pre>
<p>Transaction</p>	<p>A transaction is a Sub-Process that is supported by a special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled. 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>	
<p>Nested/Embedded Sub-Process (Inline Block)</p>	<p>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.</p>	<p>There is no special indicator for nested Sub- Processes</p>
<p>Group (a box around a group of objects within the same category)</p>	<p>A Group is a grouping of graphical elements that are within the same Category. This type of grouping does not affect the Sequence Flows within the Group. The Category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which Categories of objects can be visually displayed on the diagram.</p>	
<p>Off-Page Connector</p>	<p>Generally used for printing, this object will show where a Sequence Flow leaves one page and then restarts on the next page. A Link Intermediate Event can be used as an Off-Page Connector.</p>	

Element	Discription	Notation
Association	An Association is used to link information and Artifacts with BPMN graphical elements. Text Annotations and other Artifacts can be Associated with the graphical elements. An arrowhead on the Association indicates a direction of flow (e.g., data), when appropriate.	
Text Annotation (attached with an Association)	Text Annotations are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram.	
Pool	A Pool is the graphical representation of a Participant in a Collaboration. It 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. A Pool MAY have internal details, in the form of the Process that will be executed. Or a Pool MAY have no internal details, i.e., it can be a "black box."	
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.	

# Event Definitions

(derived from section 10.4.5 of the BPMN Specification v2.0 available at: [www.omg.org/spec/BPMN/2.0](http://www.omg.org/spec/BPMN/2.0))

Types	Start			Intermediate				End
	Top-Level	Event Sub-Process Interrupting	Event Sub-Process Non-Interrupting	Catching	Boundary Interrupting	Boundary Non-Interrupting	Throwing	
None								
Message								
Timer								
Error								
Escalation								
Cancel								
Compensation								
Conditional								
Link								
Signal								
Terminate								
Multiple								
Parallel Multiple								