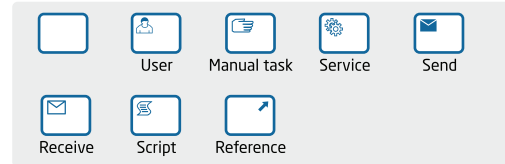


Activities [Rounded rectangles]

Activities represent the work performed by an organization; it is a step within the process. Activities can be atomic or compound.

Task

A task is a simple activity which is used when the work performed within the process is not defined at a more detailed level. BPMN defines different types of tasks:



Sub-process

Is a compound activity whose detail is defined as a flow of other activities.

Embedded sub-process
Depends completely on the parent process. It cannot contain pools or lanes

Reusable sub-process
Is a defined process like another business process diagram, that does not depend on the parent process.

Gateways [diamonds]

Gateways are elements used to control divergence and convergence of the flow. (Split and Merge)

Data-Based Exclusive Gateway
Divergence: the Exclusive Decision has two or more outgoing Sequence Flows, but only one of them can be taken and the decision will be taken after evaluating a business condition.
Convergence: is used to merge alternative paths.

Event-Based Exclusive Gateway
Is used as a Divergence element, This gateway represents a point in the process where only one of many paths of the process can be selected but based on an event, not a data expression condition.

Parallel Gateway
Divergence: is used to create parallel flow.
Convergence: is used to synchronize multiple parallel paths into one. The flow continuous when all the incoming sequence flows have reached the gateway.

Inclusive Gateway
Divergence: indicates that one or more routes can be activated from many available, and the decision is based on process data.
Convergence: indicates that many outgoing routes of an Inclusive gateway, used as an element of divergence, can be synchronized into just one.

Complex Gateway
Divergence: is used to control complex decision points that are not easy to manage with other types of gateways.
Convergence: When the Gateway is used as a Merge then there will be an expression that will determine which of the incoming Sequence Flow will be required for the Process to continue.

Events [circles]

Events represent something that happens or may happen during the course of a process. These Events affect the flow of the Process and usually have a cause or an impact and there are 3 types of events based on how the process flow is affected.

Start Events

- Indicate the instance or initiation of a process
- These do not have any incoming Sequence Flow

Intermediate Events

- Intermediate Events indicate something that occurs or may occur during the course of the process, between Start and End.
- These can be used within the sequence flow or attached to the boundary of an activity. Intermediate Events can be used to catch or to throw the event trigger.
- When the event is used to catch the Event marker will be unfilled, and when the event is used to throw the Event marker will be filled.

End Events

- End Event indicates where a process will end.
- A process can have more than one end. It does not have outgoing sequence flows.

None Start Event
Does not specify any particular behavior. It is also used for a Sub-Process.

None Intermediate Event
Indicates that something that occurs or can occur within the process. It can only be used within the sequential flow of the process.

None End Event
Indicates that a route of the process has reached its end. A process can only finish when all the routes of the flow arrive at an end

Message Start Event
A process starts when a message is received from another participant.

Message Intermediate Event
Indicates that a message can be sent or received. If the event is of reception, it indicates that the process has to wait until the message has been received. This type of event can be used within the sequential flow or attached to boundary of an activity to indicate an exception flow.

Message End Event
Indicates that a message is sent to another process when the process arrives at the end.

Timer Start Event
Indicates that a process starts at certain time or on a specified date

Timer Intermediate Event
Indicates a waiting time within the process. This type of event can be used within the sequential flow indicating a waiting time between the activities or attached to boundary of an activity to indicate an exception flow when a time-out occurs.

Conditional Start Event
A process starts when a business condition becomes true.

Conditional Intermediate Event
Is used when the flow needs to wait for a business condition to be fulfilled. It can be used within the sequential flow indicating that it should wait until a business condition has been fulfilled or attached to boundary of an activity indicating an exception flow that is activated when the condition is met.

Signal Start Event
A process starts when a signal coming from another process is captured. Note that the signal is not a message; messages have clearly defined who sent them and who receives them.

Signal Intermediate Event
Is used to send or receive signals. If it is diagrammed within the sequential flow of a process it can send or receive signals. If it is diagrammed attached to boundary of an activity, it can only receive signals and indicating an exception flow that is activated when the signal is captured.

Signal End Event
Indicates that a signal is generated when the process ends.

Multiple Start Event
Indicates that there are many ways to start the process. Only one of them will be required to start the process.

Multiple Intermediate Event
This means that there are multiple triggers assigned to the Event.

Multiple End Event
Indicates that many results can be given at the end of the process. All the results should occur.

Cancel Intermediate Event
Is only used in Transaction Sub-Process. This event is always diagrammed attached to boundary of the transactional sub-process and indicates an alternative flow that can be made when the transaction sub-process is cancelled.

Cancel End Event
Is only used in Transaction Sub-Process and indicates that the Transaction should be cancelled.

Error Intermediate Event
Is used to capture errors and to handle them. This event can only be attached to the boundary of an activity.

Error End Event
Indicates that a named Error is generated when the process ends.

Compensation Intermediate Event
The Compensation Intermediate Event enables you to handle compensations. When used within the sequential flow of a process they indicate that a compensation is necessary (throwing). When used on the borders of an activity it indicates that this activity will be compensated when the event is triggered (catching).

Compensation End Event
Indicates that the process has finished and that a compensation is necessary.

Link Intermediate Event
Is used to connect two sections of the process.

Terminate End Event
This event ends the process immediately. When one of the routes of the flow arrives at its end, indicating that the process has completely finished.

Swimlanes

Pool

- A pool is a container of a single process.
- The name of the pool can be considered as the name of the process.
- There is always at least one Pool.

Lane

- A lane is a subdivision of a pool
- Represents a role or an organizational area.

Connecting Objects

Sequence Flow

- Is used to show the order that activities will be performed in a Process.
- It is used to represent the sequence of the flow objects, where we find activities, gateways and events.



Message Flow

- A Message Flow is used to show the flow of messages between two entities or processes.
- Message flows represent messages, not flow controls.
- Not all message flows are fulfilled for each instance of the process nor is there a specific order for the messages.

Association

- An Association is used to associate information and
- Artifacts with Flow Objects.

Artifacts

Allow or provide additional information about a process.

Annotation

- Provides additional information about the process for the reader.

Group

- Is a visual mechanism that allows the grouping of activities for the purpose of documentation or analysis.

Data Object

- Provides information about the entrance and exit of an activity.