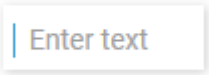
Supported BPMN Elements

The following BPMN elements are supported for modeling in the Designer:

Page	Element	Description
End Event		The **End Event** marks the end of a process. If the process consists of several branches, all other open branches continue to run to their end. Every process branch needs its proper end event.
Exclusive Gateway		Use the **Exclusive Gateway** to model a decision in the process. The process will continue with the sequence flow where the defined condition is **true**. You can also define a default sequence flow (see **Particularity**). • If you use an exclusive gateway, only one sequence flow is selected. • If no condition evaluates to **true** and you have not defined a default flow, the xUML Runtime will throw an exception. We recommend avoiding this scenario.
Free Text		The **Free Text** element allows the user to create an additional text field on the canvas, e.g. to display additional descriptions or notes.
Image (BPMN)		The **Image** element enables the user to display a picture on the canvas, for example for documentation purposes.
Lane		Use the lane element to design role-based processes. Assign one or multiple roles to a lane element and model all the steps that the assigned role(s) has/have to perform within that lane.

Message Event

The **Message Event** waits for a message with a defined content. You can use the message event in the Designer as an intermediate event or as a boundary event:

- **Intermediate:** You can use the object as an intermediate catching event. In this case, the event is a separate process step where the process must wait for a specific trigger.
- **Boundary:** The signal is associated with an activity. The event listens to a signal being fired while the associated activity is active.

Plain Event, Message Event and Timer Event can be used as **boundary** events along with User Task and Receive Task. When using the events as boundary events, attach the element directly to the border of the corresponding task:

Message Start Event

With a **Message Start Event** you can start a BPMN model. "Message" is not restricted to emails or calls: Every action that represents or contains information for a recipient is a message.

Parallel Gateway

Use the **Parallel Gateway** to model concurrency in a process. You can use the object to fork the process. The functionality of the object is based on the incoming and outgoing process flows.

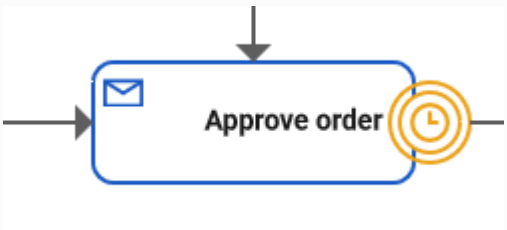
- **Forking the process flow:** If you use the parallel gateway to fork the process, all outgoing process flows are followed in parallel.
- **Joining process flows:** You can also use the parallel gateway to join several process flows which means that the process flow is only continued if all previous flows have arrived at the gateway.

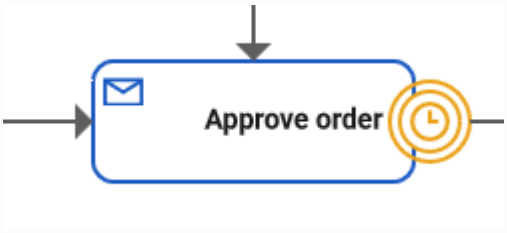
Plain Event

The plain **Event** is able to catch signals. You can use the event in the Designer as an intermediate event or as a boundary event:

- **Intermediate:** You can use the object as an intermediate catching event. In this case, the event is a separate process step where the process must wait for a specific trigger.
- **Boundary:** The signal is associated with an activity. The event listens to a signal being fired while the associated activity is active.

Plain Event, Message Event and Timer Event can be used as **boundary** events along with User Task and Receive Task. When using the events as boundary events, attach the element directly to the border of the corresponding task:



Receive Task		A **Receive Task** waits for an external trigger to arrive which is released by an external system. If the process execution reaches a receive task, the process stays in a wait state until a specific message is received by the engine, which triggers continuation of the process.
Relation		The relation connects two objects. The direction of the arrow marks the process flow. Most relations are of regular type with the exception of outgoing flows of Exclusive Gateways. For these you have two options to define the type of the sequence flow in the attributes panel: • **Regular** • **Default sequence flow**
Service Task		A **Service Task** is executed automatically when the process reaches it and the process continues to the next element when the execution ends. Use the execution pane to create the corresponding execution diagram defining the service task.
Start Event		Use a **Start Event** to start a BPMN model.
Timer Event		**Timer Events** are triggered by a defined timer. You can use the timer event in the Designer as an intermediate event or as a boundary event: • **Intermediate:** You can use the object as an intermediate catching event. In this case, the event is a separate process step. When the execution arrives at the event, a timer is started. When the timer fires, the sequence flow exiting the timer is followed. • **Boundary:** The timer is attached to an activity and works as a stopwatch and an alarm clock. When the execution reaches the activity, the timer is started. When the timer fires, for example after a defined interval, the activity is interrupted and the sequence flow exiting the timer is followed. Plain Event, Message Event and Timer Event can be used as **boundary** events along with User Task and Receive Task. When using the events as boundary events, attach the element directly to the border of the corresponding task:  To set the duration of the timer, the event should be trigged by a persisted property of type **integer** or **datetime**. You have two options to set the timeout in the attributes panel: • **Integer:** Define the number of seconds the event has to wait. • **Datetime:** Define the absolute datetime when the event should fire.

Timer Start Event		A **Timer Start Event** allows you to start a BPMN model after a defined cycle or at a defined time. Every time the timer is triggered, the model will be started.
User Task		A **User Task** waits for an external trigger to arrive which is released by human action, for example after a form has been filled. If the process execution reaches a user task, the process stays in a wait state until a specific message is received by the engine, which triggers continuation of the process.