



Events represent a milestone in a process and don't usually have a duration or associated resources. Events start, stop, delay, or interrupt process flows.

There are dozens of event variants. However, this lesson will focus on the basics.

Events are represented by circles. A start event triggers a process. A process diagram may have multiple start events. However, only the sequence flow originating from the appropriate start event receives a token.

End events consume tokens. A process diagram can have multiple end events. The process instance is completed when all tokens have been consumed.

As the name suggests, intermediate events influence the flow between the start and end events. In this example, the process flow is paused until the "Supervisor approval received" intermediate event occurs. Only then can the process flow proceed to the "Submit order" task.

Events can be further categorized as catching or throwing. Catching events are used to start an activity. When a token reaches a catching event, the token waits until the event occurs before proceeding. In this example, the "Confirmation received" event must occur before the "Fulfill order" task starts.

Throwing events are used to signal the completion of an activity. When a token reaches a throwing event, "Receipt sent," in this example, the event is executed immediately.

These are a few of the most common ways catching events can be triggered. Message events are triggered when a message is received from an external actor. In this example, the order fulfillment process is instantiated when the "Order received" message event is triggered.

Timer events are triggered at specific points in time, such as 12:00 PM on December 24th, a recurring time, such as every Saturday, or a relative time, such as 5 days after the completion of the previous activity. For example, this process is started every Monday.

Conditional events are triggered when specified criteria are fulfilled. An earlier example featured a conditional intermediate catching event, "Supervisor approval received," which pauses the process flow until the event occurs.

Here are a few common types of throwing event triggers. Message-throwing events send a message to an external actor. Here's an example of a message-throwing end event, "Newsletter distributed."

Escalation and error events are used to handle exceptions by initiating alternative sequence flows. Consider this order fulfillment process. If the product ordered is available for shipment within two days, the process flows along the happy path.

However, if the product is available but won't be ready to ship for two or more days, an escalation event is thrown. The escalation event triggers the "Inform customer" task.



Meanwhile, the rest of the fulfillment path is completed when the product is available to ship to the customer.

Sometimes, though, a product is no longer available, which triggers an error event. In that case, the customer is informed, and the product is removed from the catalog.

Finally, terminate end events immediately cancel all activities, switch off all active tokens, and stop the entire process.