BPMN EVENTS - BASICS

Events come in 3 types:

- 1. **Start events** are shown as **circles with single thin border**.
- 2. **Intermediate events** show what can happen in the course of the process and allow you to document how will it be handled. There are two types of intermediate events: the ones that are part of the process flow (think of milestones) and the ones that are attached to the boundaries of activities (think of special events that need some handling) They are **circles with double thin border**.
- 3. **End events** show what are the results/outcomes of the process, so that you know its scope. They are **circles with single thick border**.

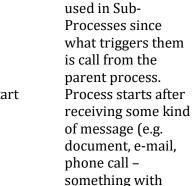


START EVENTS

Start events show what needs to happen, so that a process can start. This is NOT what you do (not activity from your side), but what triggers process to start. They are **circles with single thin border**. If they have some specific trigger defined they will have some marker inside.

Symbol	
)

Name Start event (Trigger: None)



Meaning

Name of this object

makes the process start. This type of start event is also

specific recipient)

participant (=pool)

from other

shows you what







Timer Start event

Process starts on a specific moment of time (e.g. every Monday on 9 AM, two days after quarter is closed etc.)



Conditional Start event etc.)
Process starts when certain condition is met (e.g. stock level in warehouse drops below minimum)
Process starts upon receiving some kind of signal (e.g.



Signal Start event

of signal (e.g. information about state of other process – as opposed to message there is no specifically defined



Multiple Start event

recipient)
There are several
ways in which
process can be
triggered and any
one of them is
sufficient to make it
start (e.g. process





(Multiple) Parallel Start event starts every Tuesday morning or when CEO asks for it). There are several things that need to happen together in order to trigger a process (e.g. process starts only when customer placed an order for item that we don't have in stock and due to bank holidays we cannot check with our suppliers immediately).



START EVENTS - EVENT SUB-PROCESSES

Start events can also be placed in **Event Sub-Processes**. This is not very commonly used, but very powerful concept allowing you to show how to handle special events (e.g. errors) in your process.

Apart from the markers inside you will notice that they have either solid or dashed line. The former signifies that they will interrupt the main process to do their job. The latter means they will do their job without interrupting the main process.



Message Interrupting Start event



Message Non-Interrupting Start event If a specific message is received, course of action defined in Event-Sub-Process is executed and the main process is interrupted If a specific message is received, course of action defined in Event-Sub-Process is executed without interrupting the main process.





Timer Interrupting Start event



Timer Non-Interrupting Start event



Error Interrupting Start event



Escalation Interrupting Start event

of time course of action defined in **Event-Sub-Process** is executed and the main process is interrupted On a given moment of time course of action defined in Event-Sub-Process is executed without interrupting the main process. If an error occurs, course of action defined in Event-Sub-Process is executed and the main process is interrupted. If a specific situation (not being an error) that requires process to react occurs, course of action defined in Event-Sub-Process is executed and the

On a given moment





Escalation Non-Interrupting Start event



Compensation Interrupting Start event



Conditional Interrupting Start event main process is interrupted. If a specific situation (not being an error) that requires process to react occurs, course of action defined in **Event-Sub-Process** is executed without interrupting the main process. If a need to undo steps already executed in a process occurs, course of action defined in Event-Sub-Process is executed and the main process is interrupted. If a specific condition is met. course of action defined in Event-Sub-Process is executed and the main process is





Conditional Non-Interrupting Start event If a specific condition is met. course of action defined in Event-Sub-Process is executed without interrupting the main process. If a specific broadcast (e.g. information about state of other process - as opposed to message there is no specifically defined recipient) is received, course of action defined in Event-Sub-Process is executed and the main process is interrupted If a specific

broadcast (e.g. information about

state of other process – as

interrupted



Signal Interrupting Start event



Signal Non-Interrupting Start event





Multiple Interrupting Start event



Multiple Non-Interrupting Start event there is no specifically defined recipient) is received, course of action defined in Event-Sub-Process is executed without interrupting the main process. There are several events that we are expecting and when one of them takes place, course of action defined in Event-Sub-Process is executed and the main process is interrupted There are several events that we are expecting and when one of them takes place, course of action defined in Event-Sub-Process is executed without interrupting the

opposed to message





(Multiple)
Parallel
Interrupting
Start event



(Multiple)
Parallel
NonInterrupting
Start event

main process. There are several events that we are expecting and when all of them take place, course of action defined in Event-Sub-Process is executed and the main process is interrupted There are several events that we are expecting and when all of them take place, course of action defined in **Event-Sub-Process** is executed without interrupting the main process.



INTERMEDIATE EVENTS

Intermediate events show what can happen in the course of the process and allow you to document how will it be handled.

There are two types of intermediate events: the ones that are part of the sequence flow (think of milestones) and the ones that are attached to the boundaries of activities (think of special events that need some handling). They are **circles with double thin border**.

For both of them there are further sub-types. Additionally you will see markers inside that show you type of the trigger.

Intermediate events - Sequence flow

Those events allow you to show important process milestones. They can be divided into active (called in BPMN throwing – they will have dark markers inside) and passive (called catching – with light markers inside).

Symbol



Name Intermediate Event Throwing (none) Meaning Your process just reached some important milestone and continues.















Intermediate Event Throwing Message Intermediate Event Catching Message Intermediate Event Throwing Signal Intermediate Event Catching Signal

Intermediate Event Catching Timer

Intermediate Event Catching Conditional Your process sends some message to the outside and continues.

Your process needs to wait for a message from the outside before it can continue.

Your process sends some signal about an important event.

Your process needs to wait for a signal about some specific event that took place somewhere else (in this or other process).

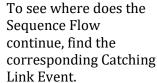
Your process needs to wait for a certain time (e.g. for 2 days or for a certain date) till it can continue.

Your process needs to wait till some external condition is met (e.g. stock level drops below some value)





Intermediate Event Throwing Link





Intermediate Event Catching Link Your process continues the flow started at a corresponding throwing link event.



Intermediate
Event
Throwing
Escalation

Your process informs higher level process (the one from which it was called) about some important event that needs to be handled in a specific way.



Intermediate Event Throwing Compensation Your process reached state where we need to undo some activities that were already executed



Intermediate
Event
Throwing
Multiple
Intermediate
Event
Catching
Multiple

Your process reached state where several events happen at the same time.



Your process needs to wait for one of several events to happen before it can continue.





Intermediate Event Catching Parallel Your process needs to wait for all of several events to happen before it can continue.

INTERMEDIATE EVENTS - BOUNDARY

Boundary events are placed on the boundary of Activities and allow you to show how will it react to a specific event. You can distinguish two types of boundary events – those that interrupt their parent when they occur (solid line) and those that do not interrupt (dashed line).

Symbol



Name Boundary Event Interrupting Message



Boundary Event Non-Interrupting Message Meaning

When a certain message is received while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event.

When a certain message is received while an Activity is being executed, additional process path (outgoing





Boundary Event Interrupting



Boundary Event Non-Interrupting Time from the event) is activated without interrupting the Activity.

When some Activity takes longer to execute than a given time or when it is still being executed when we reach some moment in time, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event.

When some Activity takes longer to execute than a given time or when it is still being executed when we reach some moment in time, additional process path (outgoing from the event) is activated without interrupting the Activity.





Boundary Event Interrupting Conditional When a certain condition turns true (e.g. it turns out we are out of stock) while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event.



Boundary Event Non-Interrupting Conditional When a certain condition turns true (e.g. it turns out we are out of stock) while an Activity is being executed, additional process path (outgoing from the event) is activated without interrupting the Activity.



Boundary Event Interrupting Signal When a certain signal is received while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow





Boundary Event Non-Interrupting Signal



Boundary Event Interrupting Escalation



Boundary Event Non-Interrupting Escalation outgoing from the event.

When a certain signal is received while an Activity is being executed, additional process path (outgoing from the event) is activated without interrupting the Activity.

When we are notified about the escalation from the lower level while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event. When we are notified about the escalation from the lower level while an Activity is being executed, additional process path (outgoing from the event) is activated





Boundary Event Interrupting Error



Boundary Event Interrupting Compensation



Boundary Event Interrupting Cancel without interrupting the Activity. When an error occurs while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event. When something goes wrong in a process, we may need to return to already completed Activities to undo them. This event is linked with association with Activities perform this undoing. You place this event on a border of a Transaction to show that it needs to be undone. It can be triggered by Cancel End Event in Event Sub-Process or specific Message.





Boundary Event Interrupting Multiple When one of several possible events happens while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event.



Boundary Event Non-Interrupting Multiple When one of several possible events happens while an Activity is being executed, additional process path (outgoing from the event) is activated without interrupting the Activity.



Boundary Event Interrupting Parallel When all of several possible events happen while an Activity is being executed, this Activity is interrupted and process continues instead with the sequence flow outgoing from the event.





Boundary Event Non-Interrupting Multiple When all of several possible events happen while an Activity is being executed, additional process path (outgoing from the event) is activated without interrupting the Activity.



END EVENTS

End events show what are the results/outcomes of the process, so that you know its scope. They are **circles with single thick border**.

Symbol



Name End Event None



End Event Message



End Event Signal



End Event Escalation Meaning

When a process reaches this end state, token is consumed. If this was the last one, process instance ends.

When a process reaches this end state, token is consumed and a message is sent.

When a process reaches this end state, token is consumed and a signal is sent.

This end event signifies that process did not complete as we would expect and we need some special handling on the higher process level.



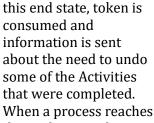


End Event Error This end event signifies that process did not complete successfully.

When a process reaches



End Event Compensation





End Event Multiple

this end state, token is consumed and several events are triggered. When Transaction Sub-

Process does not end



End Event Cancel

> successfully you want to trigger Cancel Boundary Event and send a special message that will undo content

of this Transaction.



End Event Terminate When a process reaches this end state, all remaining tokens are consumed and process instance ends.

