Creatio Academy

Sub-processes

Use sub-processes

Version 7.17

Table of Contents

Use sub-processes

PRODUCTS: ALL CREATIO PRODUCTS

Sub-processes can be used to run a process as a part of another process.

Add a sub-process

Example. A sale process (Fig. 1) must have a subordinate process for conducting a meeting with a customer (Fig. 2).

Fig. 1 Sale process

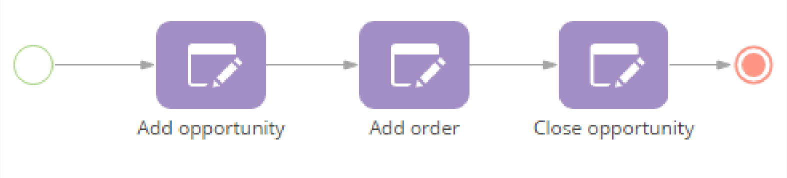
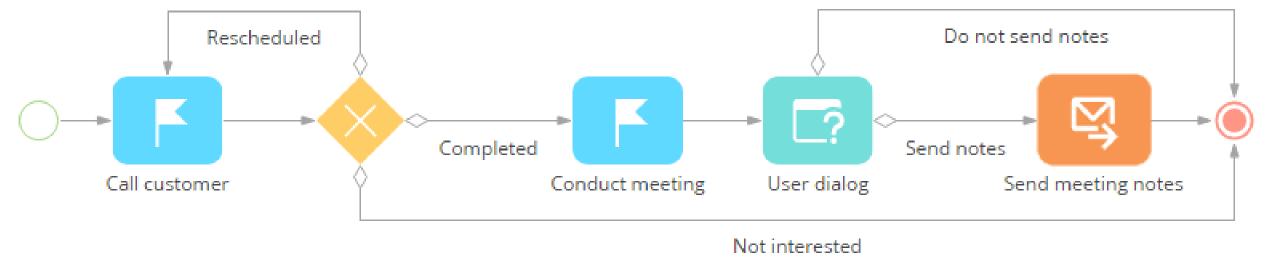


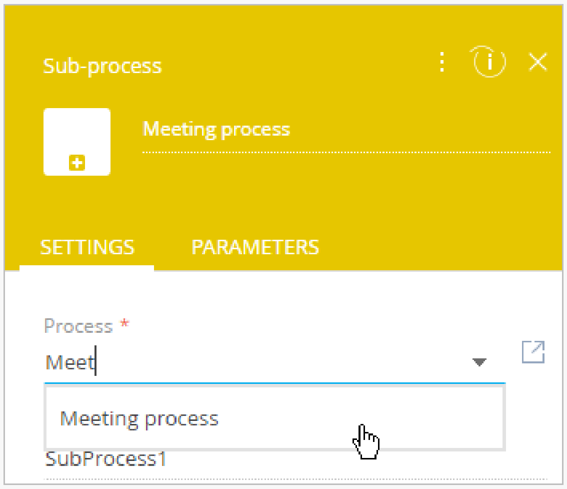
Fig. 2 Customer meeting process



Adding a meeting sub-process (Fig. 1) to the opportunity process.

1. Place the [*Sub-process*] element on the diagram after the “Add opportunity” element.
2. On the element setup area, in the [*Process*] field, select the process that must be run on executing this element (Fig. 3).

Fig. 3 Selecting a process to be run as a sub-process



Process *

Meet

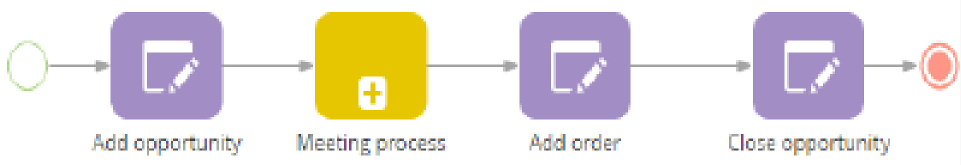
Meeting process

SubProcess1

3. Save the process diagram.

   As a result, when the parent opportunity process is run, the items preceding the sub-process will be completed first, then the sub-process (meeting with client) will be started. After the sub-process is completed, the parent process will continue (Fig. 4).

Fig. 4 Sale process with enabled meeting subprocess



To make the parent process execution depend on the subprocess result, add parameters that will be used for creating outgoing conditional flows of the sub-process element.

Obtain subprocess execution result

The parent process can pass parameter values to its sub-processes and get subprocess parameter values. Corresponding custom parameters must be added both to the parent process and the sub-process.

Example. If the meeting result is that the customer showed interest, then a new order must be created in the parent process. If the customer is not interested, then the parent process must end.

Note. The procedure for adding an order via the [*Open edit page*] element is covered in the previous article. The contract page settings are similar to those of a new document page.

To pass parameter values between the processes, first add corresponding custom parameters to both processes.

Add a parameter

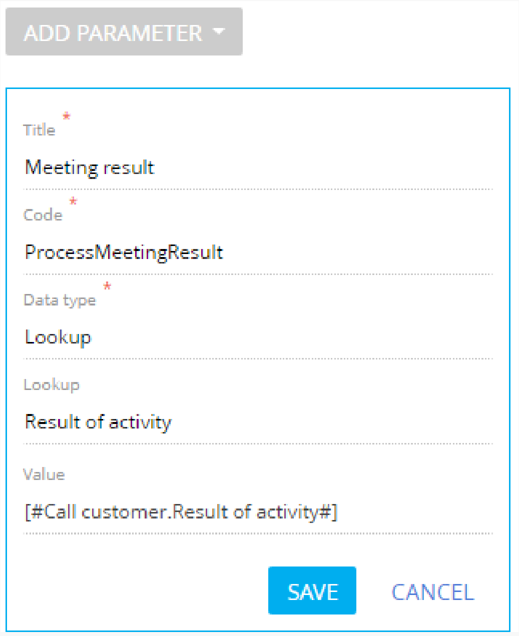
To add a parameter to the meeting process (Fig. 2)

1. Open the “Meeting with customer” process for editing and double-click on the working area of the Process Designer.

> **Note.** You can open the subprocess diagram from the [*Sub-process*] element setup area by clicking the button next to the [*Process*] field.

2. In the [*Actions*] menu of the Process Designer, select [*Process parameters*] command.
3. Click the [*Add parameter*] button and select the [*Lookup*] menu item.
4. Populate process parameter properties (Fig. 5):

Fig. 5 The properties for the process parameter



Title *
Meeting result

Code *
ProcessMeetingResult

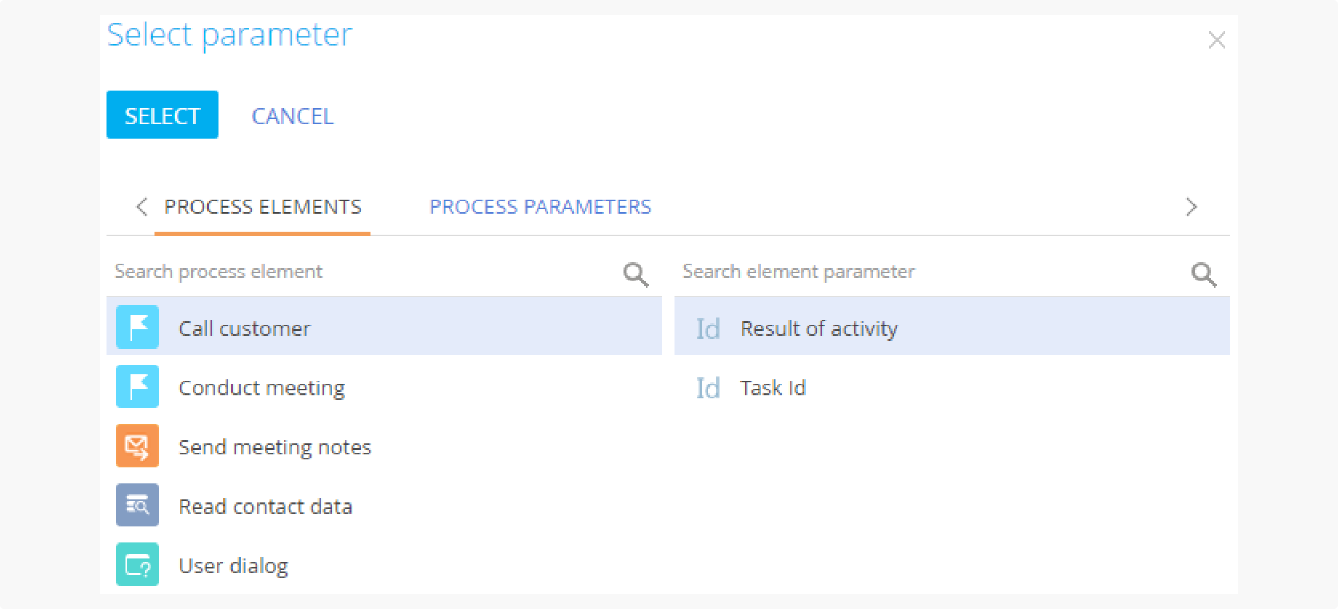
Data type *
Lookup

Lookup
Result of activity

Value
[#Call customer.Result of activity#]

1. Enter the parameter name.
2. In the [*Lookup*] field, select the [*Activity result*] lookup.
3. Specify the value source of the parameter.
   a. Click the [*Value*] field and select [*Process parameter*] in the value menu.
   b. Select the [*Activity result*] parameter of the [*Call customer*] element as a source of the process parameter value (Fig. 6).

Fig. 6 Setting up process parameter value source



Select parameter

Process element	Element parameter
Call customer	Result of activity
Conduct meeting	Task Id
Send meeting notes	
Read contact data	
User dialog	

4. Click the [*Select*] button of the parameter value window.
5. Save the process parameter.
6. Save the process.

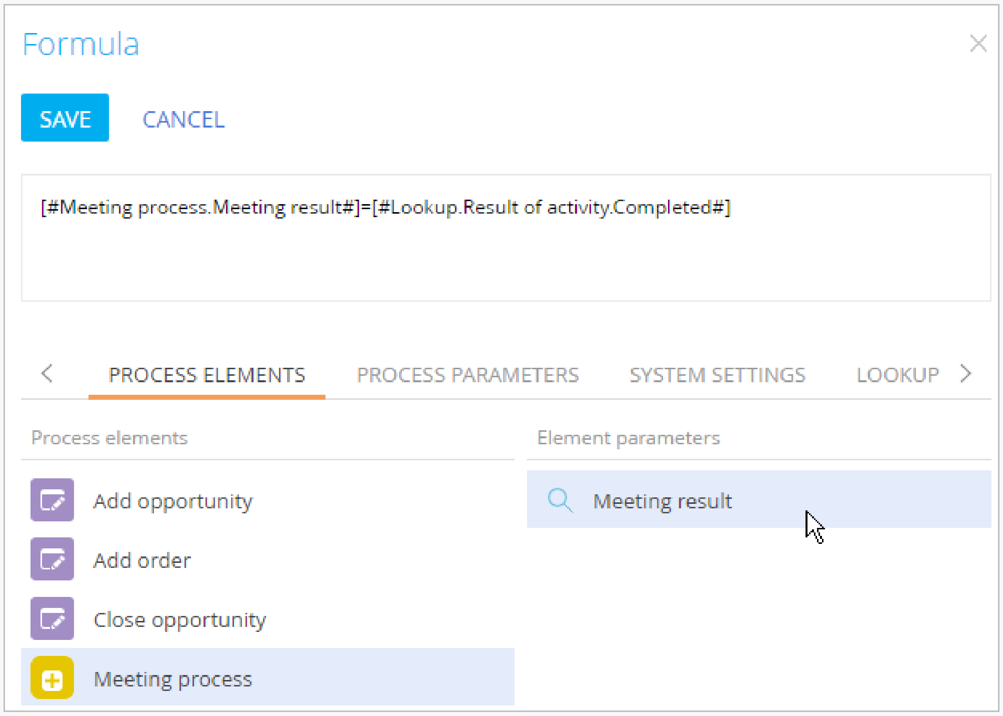
   As a result, whenever the [*Call customer*] element is completed, its result will be passed to the process parameter.

Obtain results

To specify conditions for the conditional flow that originated from the [*Sub-process*] element in the parent process (Fig. 2):

1. Open the parent process where the end parameter of another sub-process should be handled.
2. Add conditional flow from the [*Sub-process*] element to the [*Open edit page*] element that open a new document page.
3. Click the [*Condition to move down the flow*] field.
4. In the parameter value window, specify the condition for moving down the flow that compares the sub-process resulting parameter with the expected value (Fig. 7):

Fig. 7 Condition for a conditional flow

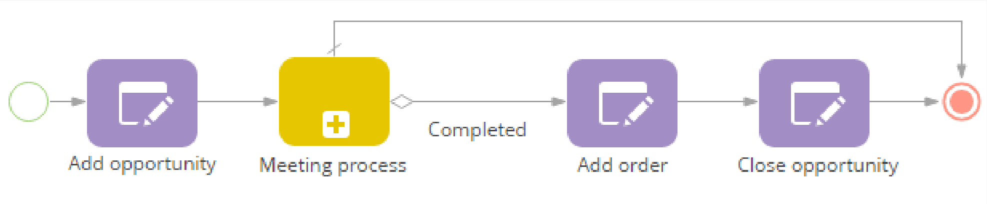


[#Meeting process.Meeting result#]=[#Lookup.Result of activity.Completed#]

a. On the [*Process elements*] tab, select [*Meeting with customer*].
b. Select the [*Conduct meeting*] process parameter.
c. In the formula area, enter “==”.
d. On the [*Lookup*] tab, select [*Result of activity*] lookup from the dropdown list.
e. Select [*Completed*].

5. Save the changes in the parameter value window.
6. Add a default flow from the subprocess element to the end process event element (Fig. 8).

Fig. 8 The meeting sub-process on the parent process diagram



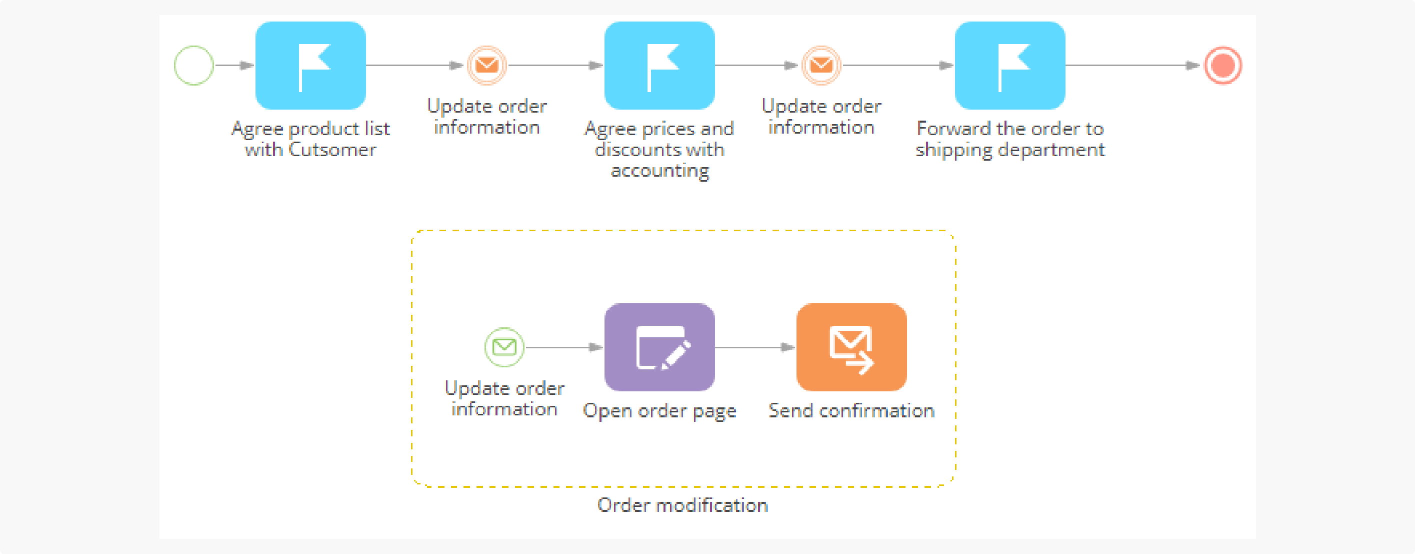
7. Save the process diagram.

Use an event sub-process

You can set up business processes containing task combinations that can be repeated several times during the process execution, when certain events occur. To arrange such task combinations in a separate process flow, use the [*Event sub-process*] element in your diagram.

For example, within a sale process, order information must be updated at every stage of negotiations (Fig. 9).

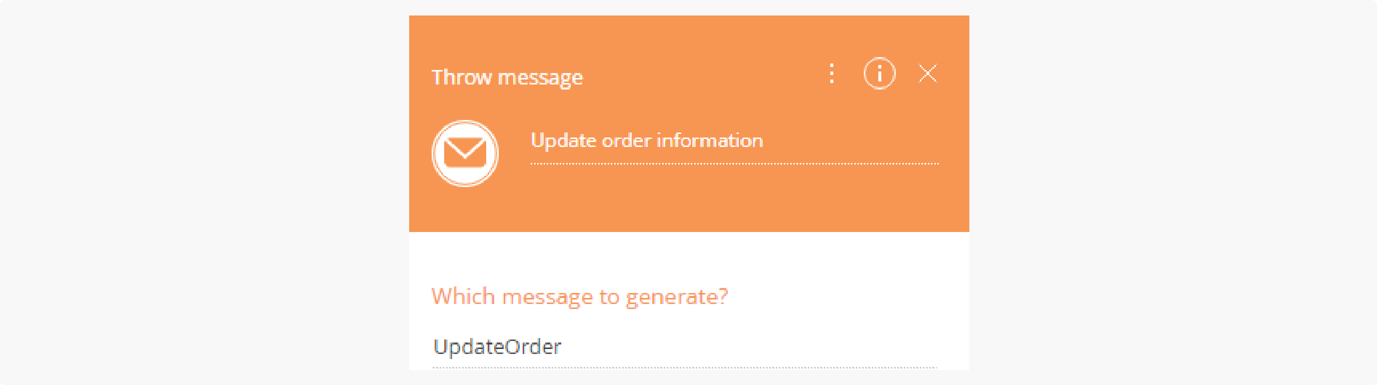
Fig. 9 Sale process containing an “Order modification” event sub-process



To set up a process containing an “Order modification” event sub-process:

1. Add a [*Throw message*] intermediate event to the process diagram after the corresponding process action. Populate the [*Which message to generate?*] field with the message name that will serve a trigger of your event sub-process, e.g. “UpdateOrder” (Fig. 10).

Fig. 10 Populating the [*Which message to generate*] field of the [*Throw message*] event in the current process



Throw message

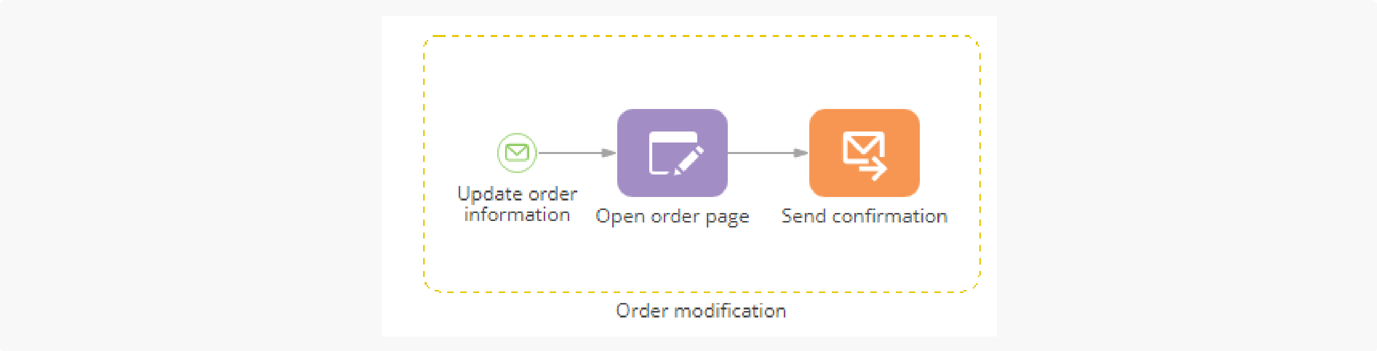
Update order information

Which message to generate?

UpdateOrder

2. Add the [*Event sub-process*] element to the diagram and set up the elements on it (Fig. 11):

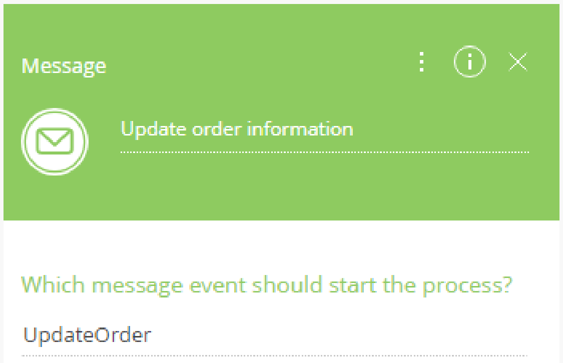
Fig. 11 [*Event sub-process*] element



1. Add a [*Message*] start event in the [*Event sub-process*] area on the diagram, as the initial event of your

event sub-process. Make sure that the value in its [*Which message event should start the process?*] field matches the value of the message generated by the corresponding [*Throw message*] event (Fig. 12).

Fig. 12 Populating the [*Which message event should start the process*] field of the [*Message*] start event in the event sub-process area



Update order information

Which message event should start the process?

UpdateOrder

a. Add subprocess steps to the diagram For example, add an [[*Open edit page*]]() element that opens the order page for the user and a [*Send email*] element to notify all involved parties that the order has been updated.

b. Connect the event sub-process elements with sequence flows.

2. Save the process diagram.

> **Note.** The event sub-process will be triggered each time the [*Message*] start event receives a corresponding message generated during the execution of the current process. Running event sub-process will not interrupt the execution of the current process: the event sub-process elements will be performed as regular process elements in the order defined by the user.

As a result, the event sub-process will launch each time the [*Throw message*] intermediate event of the current process generates a corresponding message during the process execution. For example, an order modification sub-process will run whenever an order modification message is received within the process.