ENTERPRISE ARCHITECT

User Guide Series

BPSim Business Simulations

Author: Sparx Systems
Date: 2026-05-04
Version: 17.1

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

BPSim Business Simulations 3
- Installing BPSim 6
- BPSim Configuration 7
  - BPSim - Configure Page 9
  - BPSim - Execute Page 19
  - BPSim - Step Page 21
  - BPSim - Review Page 26
  - Using the Parameter Value Dialog 28
- Using the BPSim Execution Engine 31
- BPSim Execution Engine - Simulation Language 35
- Tracking Property Parameter Values 38
  - Tracking Property Parameter Values - Examples 40
- Compare BPSim Configurations 49
- BPSim Charts 52
- BPSim Examples 56
  - Meal Order Collaboration Version 1 57
  - Meal Order Collaboration Version 2 61
  - Help Desk Phone Support Simulation 64
  - Calendar-Based Help Desk Phone Support Simulation 73
  - Car Repair Process 77
  - BPMN2.0 Event Examples 84
    - Error Event 85
    - Escalation Event 89
    - Event Sub-Process 92
    - Fibonacci Number Generator with Link Event 97
    - Message Event 100
    - Signal Events 105
    - Timer Event - Boundary 113
    - Timer Event - Standalone Intermediate Event 116
  - Paint Wall Process Simulation (Call Activity) 119
  - BPSim Cost Parameters 125
    - Set Cost Parameters on Activity 126
    - Set Cost Parameters on Resource 130
- Export a BPSim Configuration 133

BPSim Business Simulations

The open BPSim specification provides a rich set of materials on how to configure and assign resources to activities or tasks, how to raise events, decision making and other real-world capabilities. Once configured according to the BPSim specification, a business process model (constructed in BPMN) can be passed to a suitable BPSim simulation engine and run according to the process defined in the BPMN model, using the configuration data attached in the BPSim information.

The BPSim specification is very detailed and offers the interested modeler and business strategist an unprecedented flexibility in assigning operating information to a model and then assessing the quality of the solution based on information received back from the Simulation engine. This section describes in detail the various screens and options available when configuring a model for BPSim execution.

Sparx Systems provide a BPSim-capable Simulator - the **BPSim Execution Engine.** This Add-In integrates with the BPSim and BPMN models defined in Enterprise Architect, providing the capability to run and store the results from multiple simulations and to perform convenient comparisons across each configuration's result set.

The **BPSim Execution Engine** is a pre-requisite for accessing and using the BPSim configuration facilities. The Execution Engine is integrated with the Unified and Ultimate editions of Enterprise Architect; for use in the Corporate edition, it can be purchased and installed under separate licence.

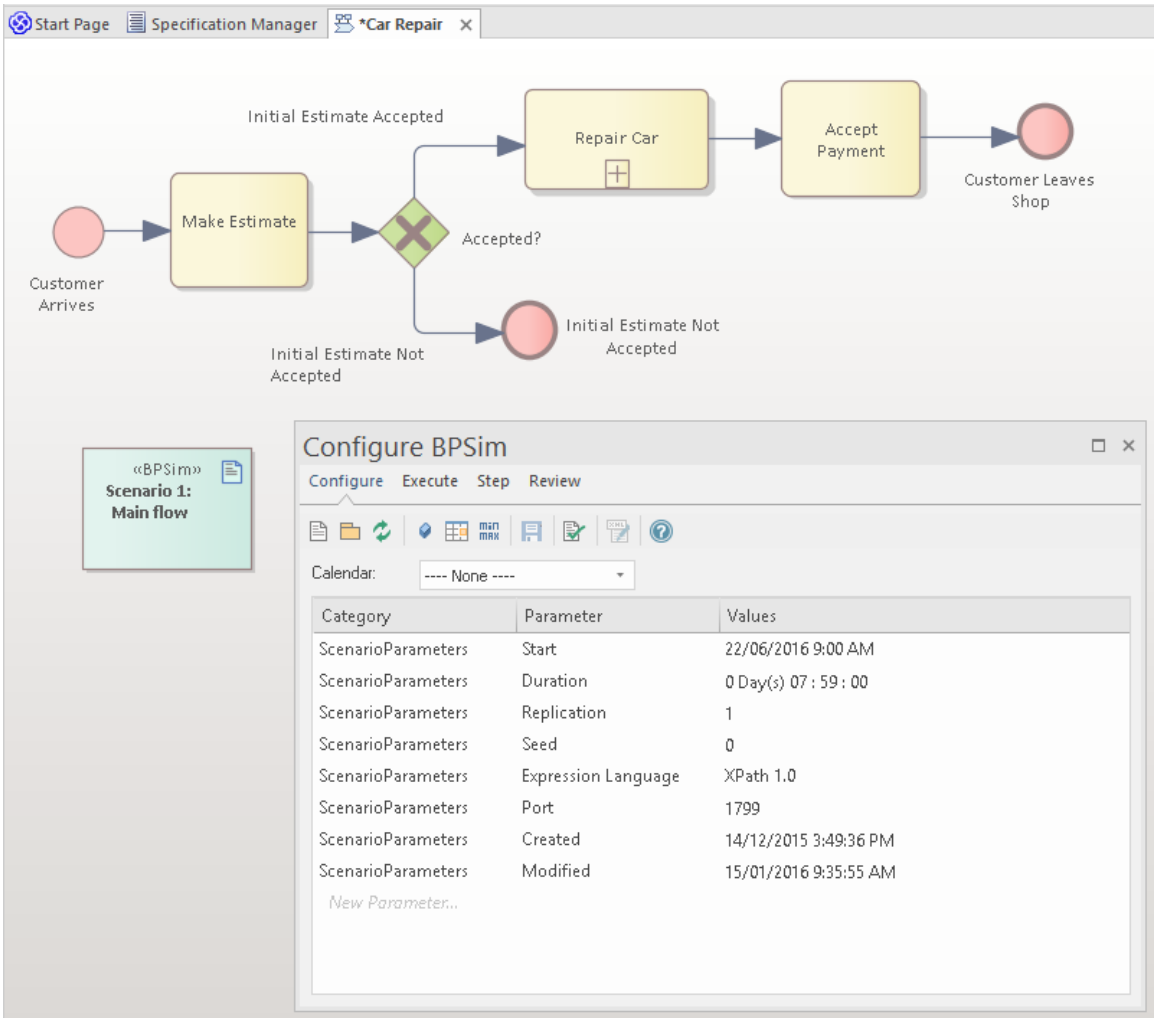
Once you have set up a BPSim configuration, the simulation execution process exports the BPMN model with its BPSim data in a standard form. This ensures that changes to the model are always incorporated into the simulation. Similarly, the model export process captures the BPMN model with its BPSim data in a form that can be imported into another model and consumed by the Sparx Systems BPSim Execution Engine or by any other standards-compliant BPSim engines.

Install BPSim

Whilst BPSim is integrated with the Unified and Ultimate editions of Enterprise Architect, it is separate from the Corporate edition and - after purchase - has to be installed on your system.

For all three editions, you must ensure that the right versions of Java Runtime Environment (JRE) and Java Development Kit (JDK) are also installed on your system.

BPMN Model With BPMN Simulation



The Configure BPSim window helps you to define several categories of Simulation parameter, each category focusing on one aspect of the Simulation Configuration. For example, you would define:

- ScenarioParameters, which define how the Simulation itself should proceed
- Control Parameters, which examine how activity flows through the business process, moderated by the likelihood of a sequence of events and the priorities of certain events
- Time (Temporal) Parameters, which examine how the duration of one or more phases in the processing of an Activity influences the business process
- Resource Parameters, which examine the involvement of types and roles of workers and other resources, their required numbers, their costs and their availability

You can also maintain multiple versions of a configuration (as separate BPSim Artifacts) and easily compare the differences between versions to see how each configuration will vary the flow of the proposed Simulation or process execution. You might, for example, establish a baseline configuration and then create multiple 'what-if?' configurations that vary one or more parameters. Once you have run these configurations through a Simulation Engine, you can examine each result and decide on the relative merits of each configuration. One useful principle to apply here is the simple inheritance of common, unchanged data in one configuration by another configuration that contains only the data being varied - you can therefore run the simulation on a current set of variables, which draws on the standard data configuration at the same time.

Users can combine the BPSim and Charting facilities to quickly vary, simulate and compare aspects of a Business Process model, and show the differences between the Simulations in one of the many Chart formats.

If you are working across multiple projects, you can export and import the BPSim configurations between them. The configuration automatically carries with it the BPMN 2.0 model on which it is based.

The Enterprise Architect Business Process Simulation configuration tool is based on the BPSim Framework developed by the Workflow Management Coalition (WfMC).

Notes

- If you click on a business process element or connector in a diagram or in the Browser window, it is highlighted and selected in the Configure BPSim window
- The Business Process that you simulate can contain elements from more than one Package; to include the external elements in the simulation, you must create a Package diagram containing the 'parent' Package and either the 'external' Packages containing the external elements, or the external elements themselves; create a:
  - Package Import connector from the parent Package to each external Package, or
  - Dependency connector from the parent Package to each external element

Installing BPSim

Whilst BPSim is integrated with the Unified and Ultimate editions of Enterprise Architect, it has to be installed on your system. For these editions, you must ensure that the right versions of Java Runtime Environment (JRE) and Java Development Kit (JDK) are also installed on your system.

Install JDE and JDK

To use the Sparx Systems BPSim Execution Engine, you must have on your system Java Runtime Environment (JRE) version 1.7 or higher and, if your BPSim Configuration contains any property parameters, you must also have Java Development Kit (JDK) version 1.7 or higher.

You do not need to do any further configuration for the engine unless you have multiple versions of JRE/JDK on your system and you want to specify which version the execution engine should use. In this case, apply these environment variables as shown:

1. Click on the Windows 'Start' icon and select the 'Computer' option.
2. From the banner menu, select the 'System properties' option.
3. From the side panel, select the 'Advanced system settings' option.
4. On the 'Advanced' tab of the 'System Properties' dialog, click on the Environment Variables button.
5. On the 'Environment Variables' dialog, in the 'System variables' panel, click on the New button.
6. On the 'New System Variable' dialog, complete the fields with the values shown:

   For JRE: Variable name: MDG_BPSIM_JRE_HOME
   Variable value: C:\Program Files\Java\jre7

   For JDK: Variable name: MDG_BPSIM_JDK_HOME
   Variable value: C:\Program Files\Java\jdk1.7.0_51
7. Click on the OK button.
8. You must re-start your machine for the new variables to take effect.

BPSim Configuration

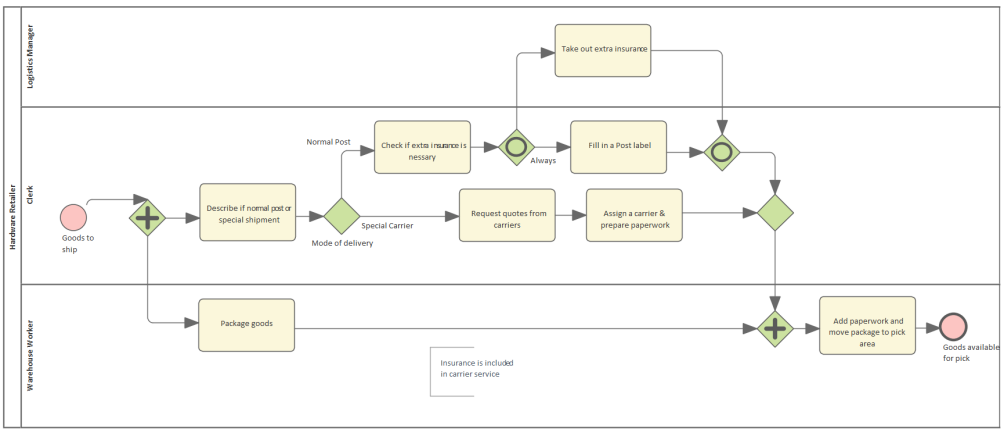
A Business Process Simulation (BPSim) configuration is represented and contained in a Business Process Simulation Artifact element, which you can create on a diagram in any Package in the same project as the BPMN model that you are working with.

Create a Business Process Model

Each BPSim configuration is created specifically for and from an existing Business Process, defined in BPMN. Therefore you will need to create or import the BPMN model on which the configuration is to be based, before you use the Business Process Simulation Artifact.

This example diagram can be found and worked on in the EAExample model, in:

Analysis and Business Modeling > BPMN 2.0 Examples > Process Diagrams > Shipment Process of a hardware retailer



Create a Business Process Simulation Artifact

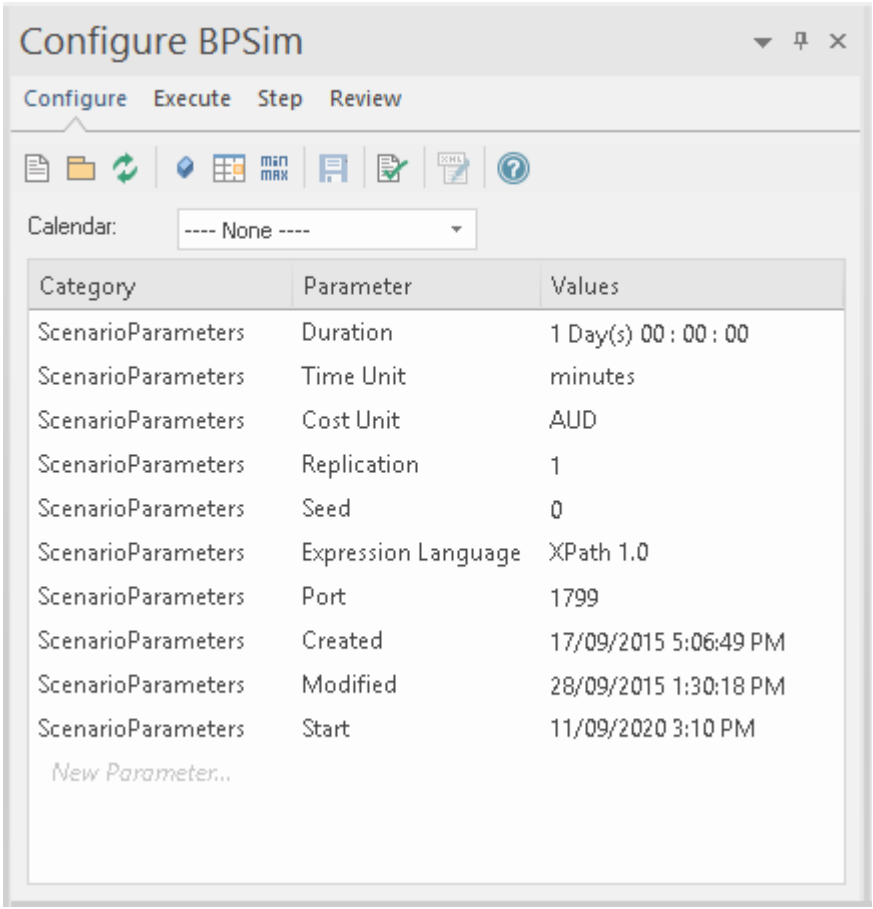
Open a diagram in which to create the Artifact, and display the Diagram Toolbox (press Ctrl+Shift+3). Expand the common 'Simulation' page, and drag the 'Business Process Simulation' icon onto the diagram.

When you set up the Artifact, consider whether you might create one Artifact to define the base configuration, and other Artifacts to define variations or additions in certain aspects of the simulation. You would use Generalization connectors between the 'base' and 'variation' Artifacts so that the variations inherit the data you have defined in the 'base' Artifact. This way, you do not have to keep re-defining the whole configuration in every Artifact you create.

Double-click on the element and give it an appropriate name, such as (for the example) 'Base BPSim Configuration'.

Configure BPSim Window Overview

Right-click on the Artifact element (either in the diagram or in the Browser window) and select the 'Configure BPSim' option. The Configure BPSim window displays for the Artifact.



This window contains four tabs: Configure, Execute, Step and Review.

- Configure: configure BPSim parameters for each BPMN element; define Property parameters, Calendars and Scenario parameters
- Execute: execute the BPMN model with a BPSim configuration
- Step: step over / step in to provide an insight into the execution process, including token status, property values and resource allocations per time/step
- Review: review / compare configuration artifact(s), generate standard or customized simulation result report(s)

BPSim - Configure Page

The BPSim Artifact will be configured to a Package. All BPMN elements under this Package or its sub-Packages will be loaded. By default, the Package containing this Artifact will be configured when loaded into this window.

This window is context sensitive. When an element is selected on a diagram or the Browser window, the list will show current configurations for the element; also, the drop-down lists of values will only show available parameters for the element.

When the BPSim Artifact is the context element, the list will show the ScenarioParameters.

Access

Ribbon	Simulate > Process Analysis > BPSIM > Open BPSIM Manager > Configure

Toolbar Options

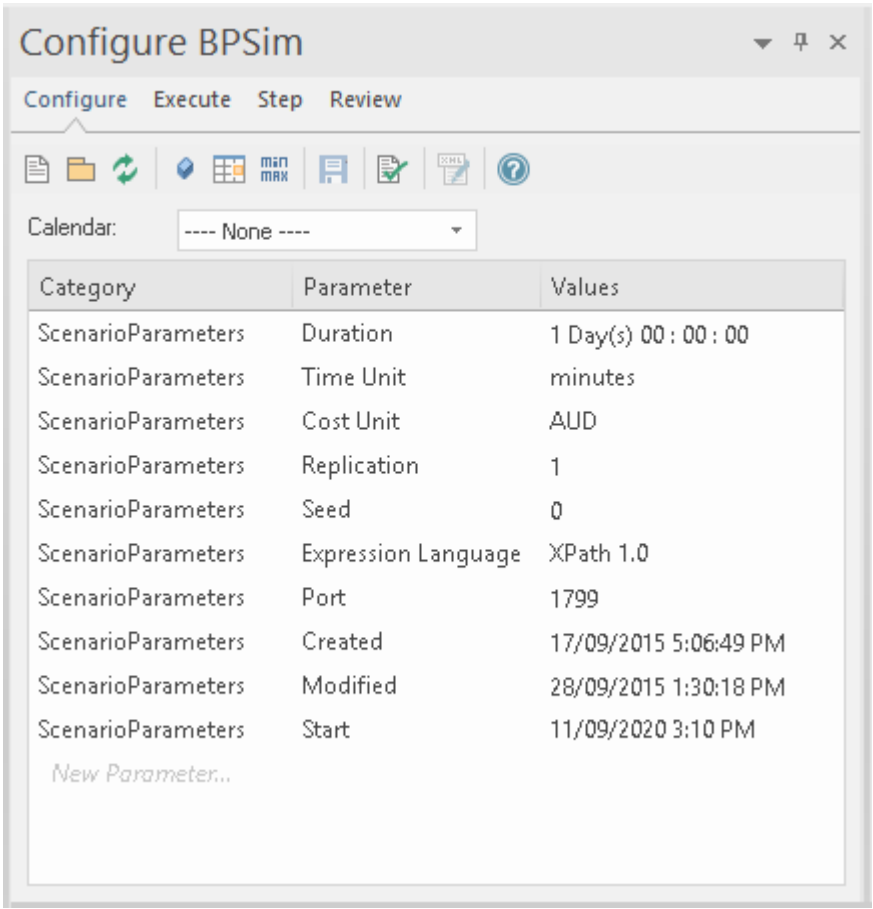
Option	Description
	Click on this button to select or create a BPSimConfiguration element.
	Click on this button to set a Package for the BPSim Artifact. All BPMN elements under this Package or its sub-Packages will be included.
	Click on this button to reload BPMN elements from the configured Packages. For example, when some BPMN elements are modified, run this command to reload the Package so that the changes will be taken into account for BPSim Simulation.
	Click on this button to define Properties, which can be used as Property Parameters on BPMN elements.
	Click on this button to define Calendars, which can be used to configure element parameters.
	Click on this button to show or hide the 'Result Request' column. Result Request configuration is required for a Custom simulation. The execution report will only contain results that are requested.
	Click on this button to save the Configure BPSim window information to a BPSim Artifact element.
	Click on this button to validate the BPMN model and the BPSim configurations. Error or warning messages might display in the System Output window if they are generated.
	Click on this button to export the BPMN model with the BPSim configuration. This exported BPMN file conforms to the BPMN and BPSim specifications and can be used by third-party BPSim Execution engines.

Scenario Parameters

A scenario is composed of a collection of element parameters. The scenario itself defines parameters used by all elements as global settings. Not all parameters will display for an element, but you can bring them in to the list by:

1. Clicking on the *New Parameter* text, clicking on the drop-down arrow and selecting 'ScenarioParameter'.
2. Clicking on the drop-down arrow in the 'Parameter' field and then selecting the parameter type from the list.

Note that once you have added all possible parameters for a scenario, the Configure BPSim window does not allow you to attempt to add more.



Name	Description
Start	The date and time at which the process starts to take effect. You can edit this by overtyping the values or, for the date, by selection from a drop-down calendar.
Duration	The length of time the process takes. The 'Duration' parameter is a required value. It must be long enough to accommodate a complete simulation; for example, if a process (and hence its simulation) takes three hours to complete, the 'Duration' parameter must be set to a value greater than three hours. You can edit this by overtyping the appropriate segment in the format 'days hours:minutes:seconds'.
Time Unit	The base unit in which periods of time are expressed in this scenario. All numeric

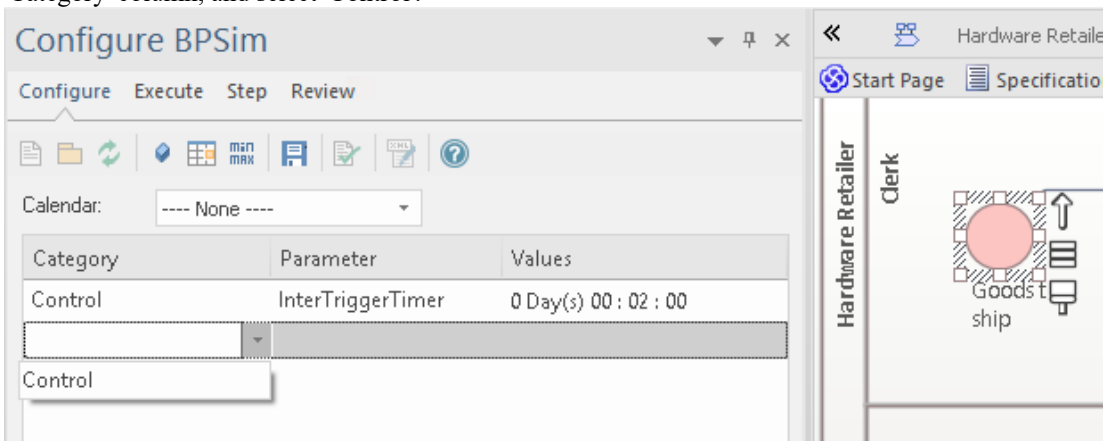
	and floating values representing time should be considered as being expressed in this unit, unless overridden locally. You can edit this by clicking on the drop-down arrow and selecting the unit.
Cost Unit	The currency unit of any costs recorded in the process. All numeric and floating values representing a cost should be considered as being expressed in that currency code, unless overridden locally. You can edit this by clicking on the drop-down arrow and selecting the unit abbreviation.
Replication	The number of replications of the scenario to be executed. Defaults to 1. You can edit this by simply typing a value in.
Seed	A random seed to be used to initialize a pseudo-random number generator. You can edit this by simply typing a value in.
Expression Language	XPath 1.0 and Java - XPath 1.0 is the default language. If Java is specified as the expression language, JDK Home must be set. You can edit this by clicking on the drop-down arrow and selecting the language.
DMN Module	When business rule tasks are used in the BPMN Model, you can implement these tasks as a DMN Model. You might first create a DMN Model and generate a DMN Server in Java, then click on the ... button to specify the generated DMN Server file.
JRE Home	The Enterprise Architect BPSim Execution Engine runs in a Java Environment, therefore a JRE Home has to be specified. Click on the ... button to choose a directory; for example, C:\Program Files\Java\jre7. You can edit this by clicking on the ... button again to browse the directory.
JDK Home	When the expression language is Java, the Enterprise Architect BPSim Execution Engine will generate Java code and compile with javac as the vendor extension. So a JDK Home must be specified. Use the ... button to choose a directory (such as C:\Program Files\Java\jdk1.7.0_80). You can edit this by again clicking on the ... button to browse the directory.
Port	The Port number that Enterprise Architect used to communicate with the BPSim Execution Engine. The default Port number is 1799.
Created	Read only field. The timestamp when the BPSim Artifact was created.
Modified	Read only field. The timestamp for when the BPSim Artifact was last modified.

Control Parameters

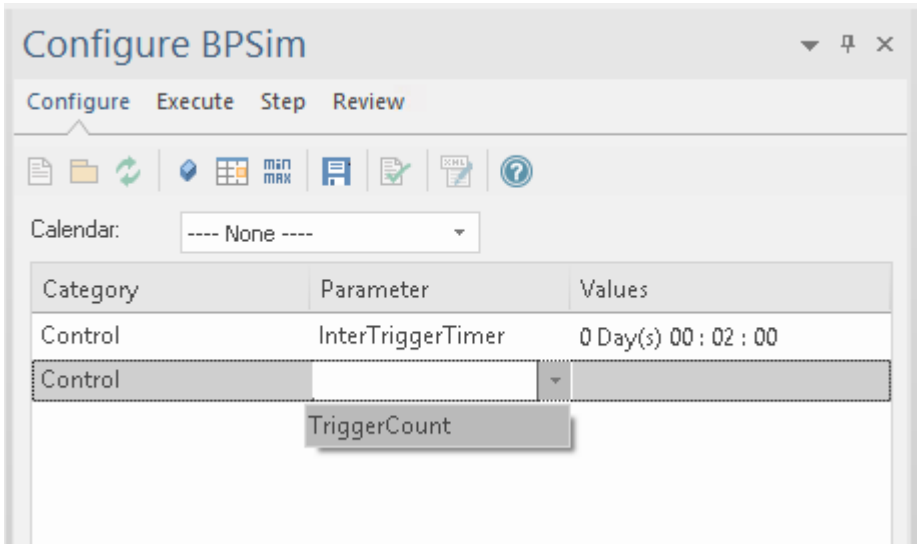
To begin defining Control parameters for the appropriate element (such as an Event or Gateway):

1. Select the element on the diagram, then click on the *New Parameter* text and on the drop-down arrow in the

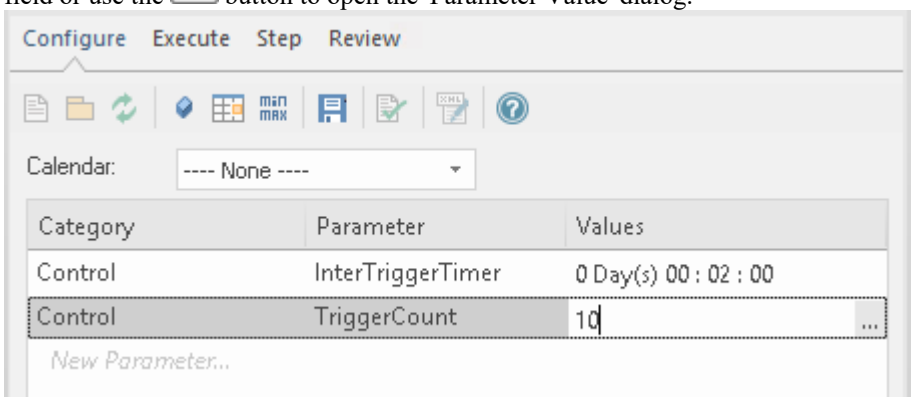
'Category' column, and select 'Control'.



2. Click on the drop-down arrow in the 'Parameter' field, which will display the available unassigned parameters for the selected element.



3. Select the appropriate parameter, and then click on the 'Values' field; you can either type the parameter value in the field or use the ... button to open the 'Parameter Value' dialog.

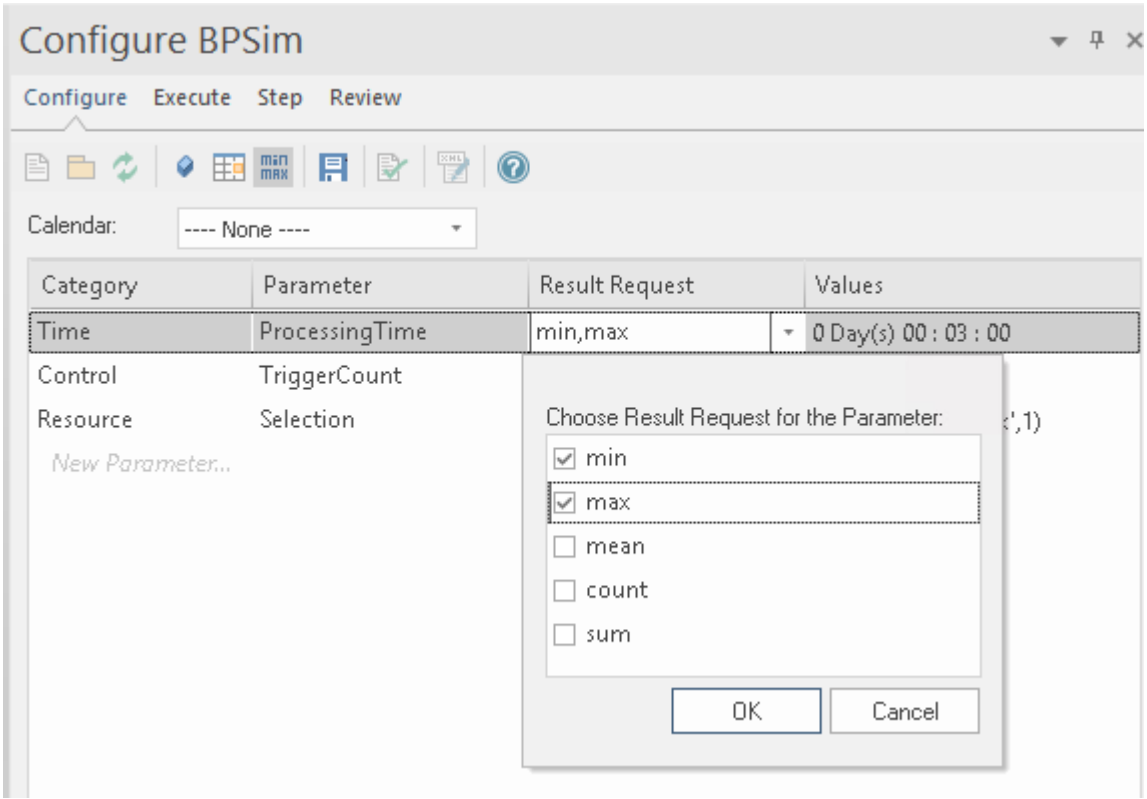


Note that the page allows you to provide only the appropriate parameters for the element. Once you have specified these parameters, the fields do not allow further input or selection.

Temporal Parameters

To begin defining Time parameters for the appropriate element (such as a BPMN Task):

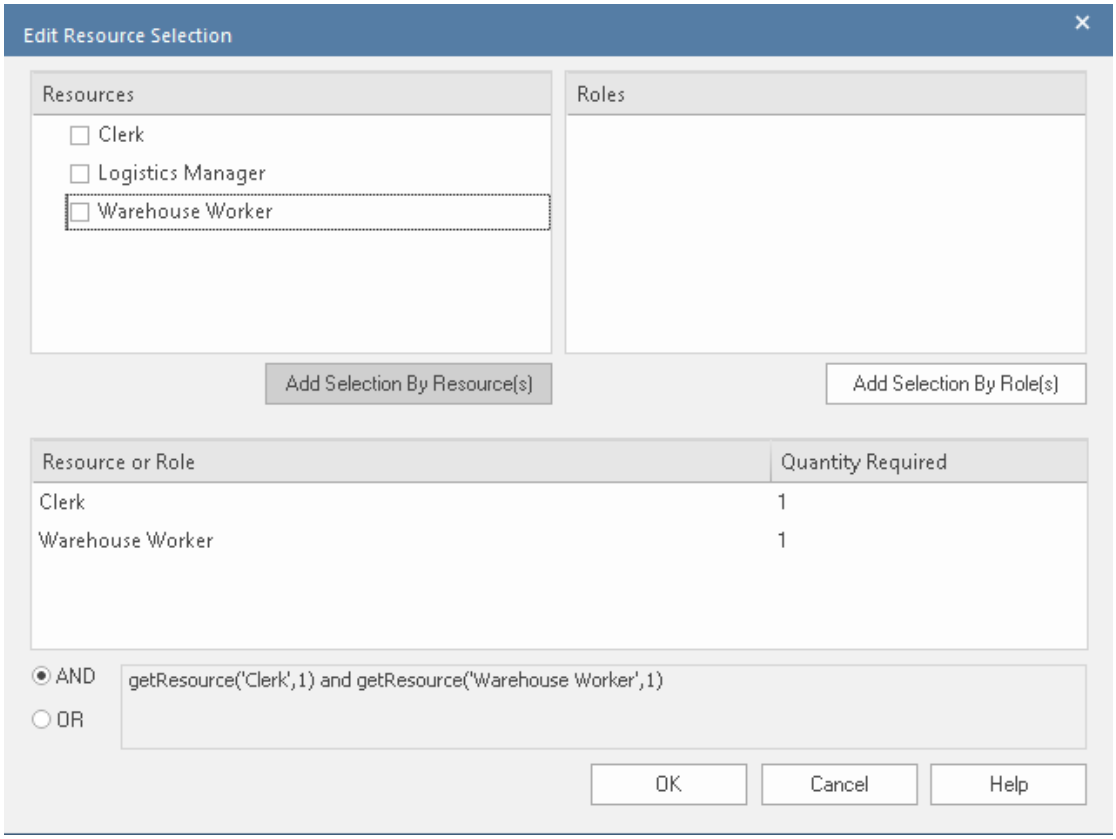
1. Select the element on the diagram.
2. Click on the *New Parameter* text and on the drop-down arrow, and select 'Time' from the list.
3. After you select 'Time', click on the drop-down arrow in the 'Parameter' field and select from the available parameters for the element.
4. In the 'Values' field, either type the value or click on the ... button to open the 'Parameter Value' dialog.
5. You can toggle the 'Result Request' column by clicking on the min/max button on the toolbar to customize the simulation output by requiring certain results



Resource Parameters

To begin defining the Resource parameters for the appropriate element (such as a BPMN Task):

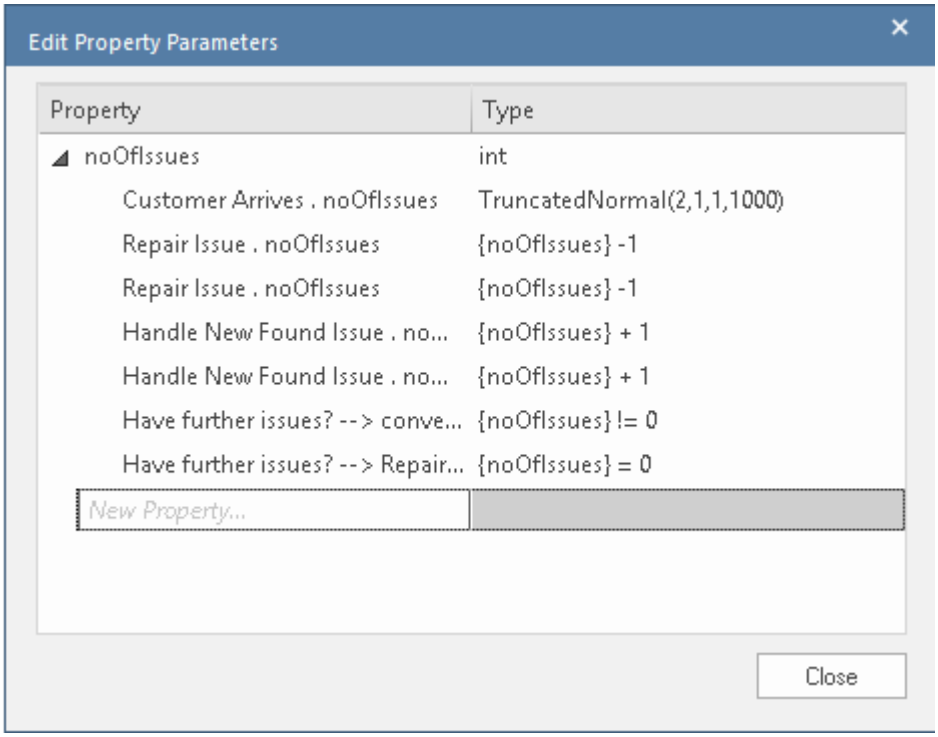
1. Select the element on the diagram.
2. Click on the *New Parameter* text and on the drop-down arrow, and select 'Resource' from the list.
3. In the 'Parameter' field, click on the drop-down arrow and click on 'Selection' in the list.
4. In the 'Values' field, click on the ... button to open the 'Edit Resource Selection' dialog.



- The top left panel lists the defined Resource elements; click on the resources to assign and on the Add Selection By Resource(s) button to move the selection to the 'Resource or Role' panel
- The top right panel lists the defined Roles (if any) for the Resource elements; click on the required roles and on the Add Selection By Role(s) button to move the selection to the 'Resource or Role' panel
- The 'Quantity Required' column defaults to 1 for each resource/role; if a larger quantity is required, overtype this value with the appropriate number
- Click on the appropriate radio button to set the logical relationship to AND or OR for the selection
- The final expression for Resource selection is composed and shown in the text field
- Click on the OK button to return to the Configure BPSim window, where the expression is shown in the 'Values' field

Property Parameters

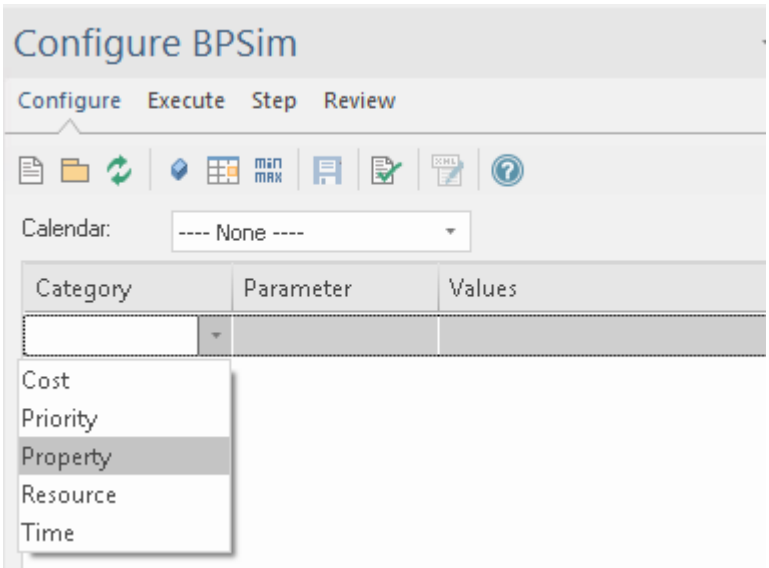
To begin defining the Property parameters, click on the button on the toolbar. The 'Edit Property Parameters' dialog displays.



The defined properties and their references are listed.

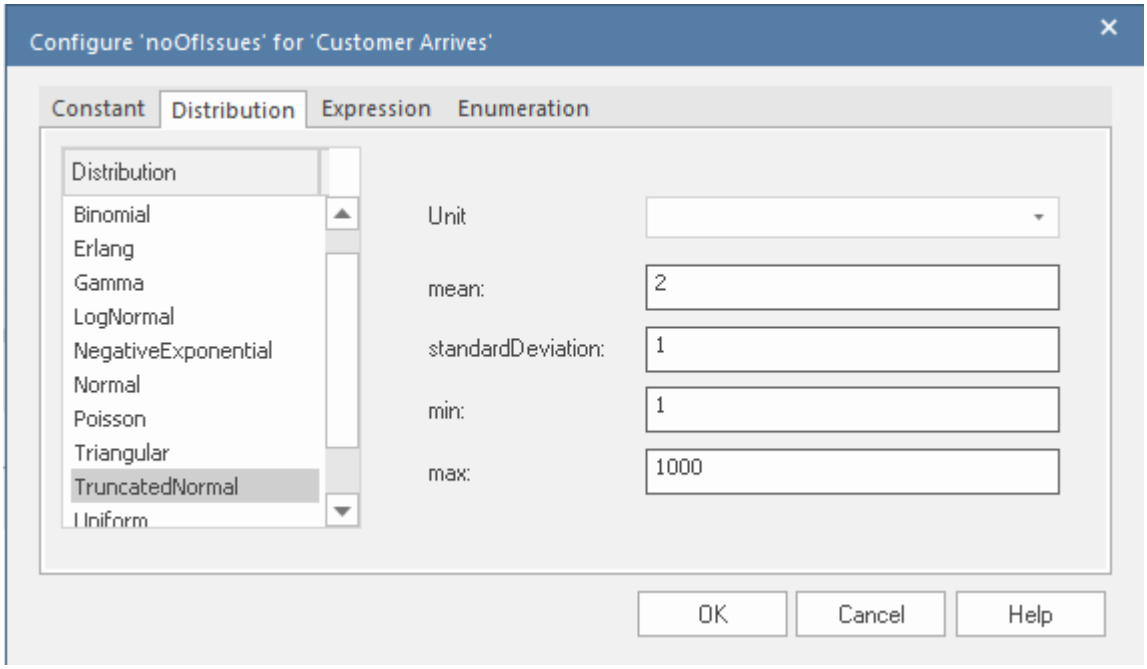
You can add a new property, delete a selected property (using the context menu option), overtype a property's name or select a different type for a property.

Having checked the defined properties, you can set Property parameters on BPMN elements.



Choose 'Property' as a category, then click on the drop-down arrow in the 'Parameter' field and select a property.

Click on the ... button on the 'Values' column to display the parameter value dialog (named from the property and the parent element).



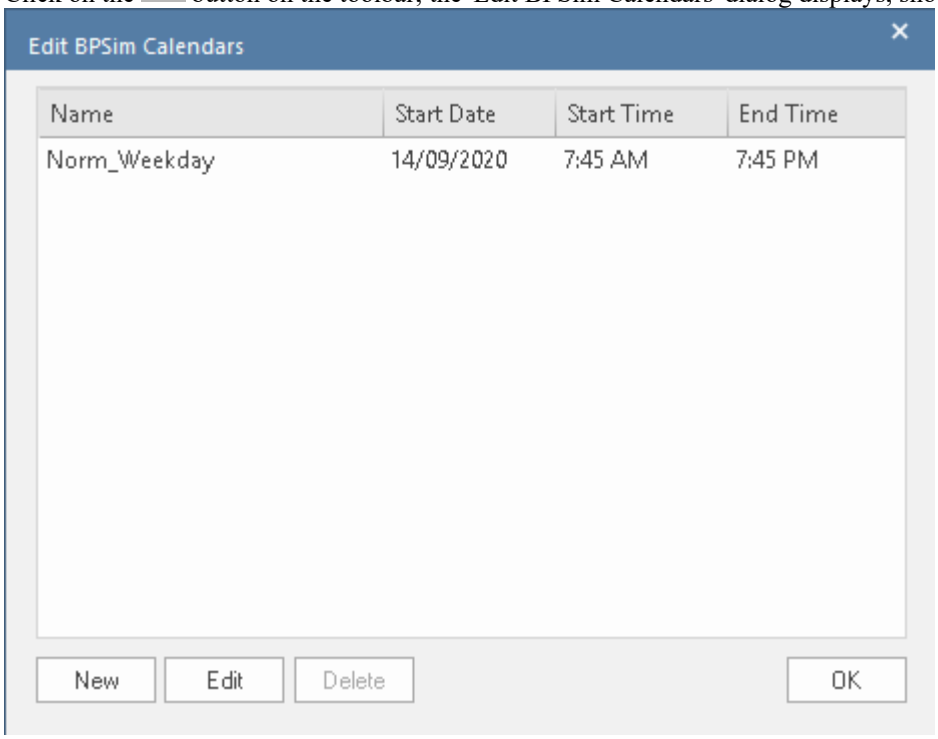
Navigate to the appropriate tab to select and define the type of value and actual value, then click on the OK button. The value displays in the 'Value' field.

Calendars

Calendars help you to define any number of special periods of time that can influence the process, such as working days, shifts, holidays or periodic events (for example, stock-taking, inventory or auditing).

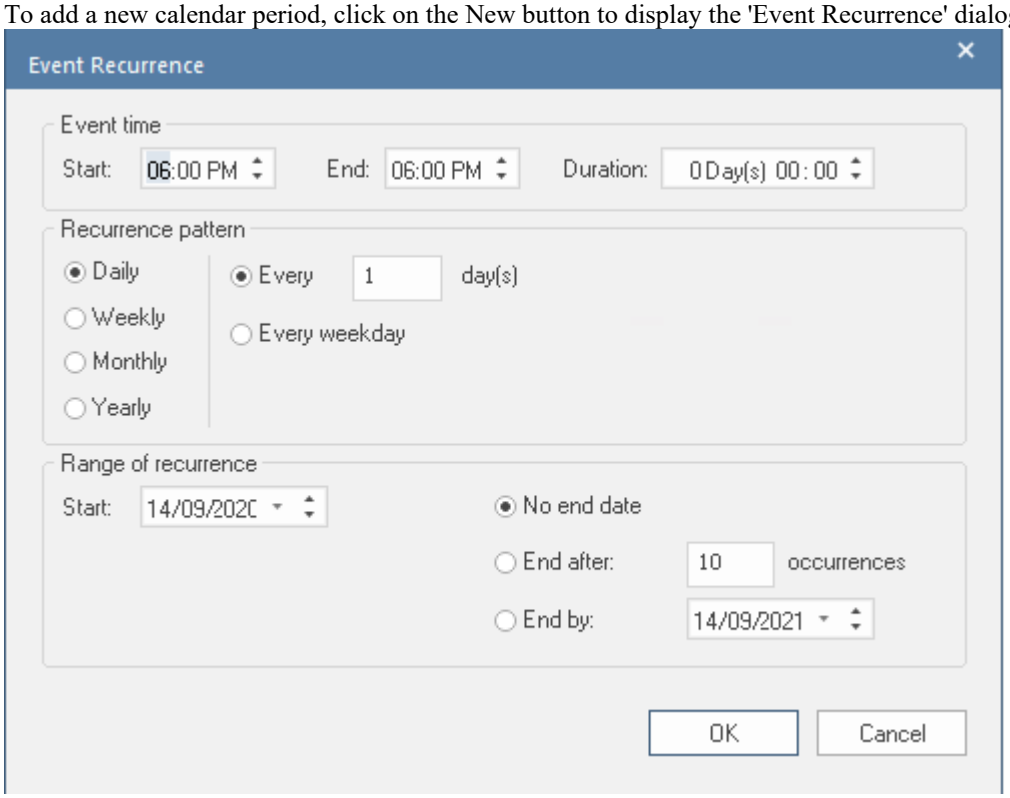
To begin defining the Calendars:

1. Click on the button on the toolbar; the 'Edit BPSim Calendars' dialog displays, showing any existing calendars.



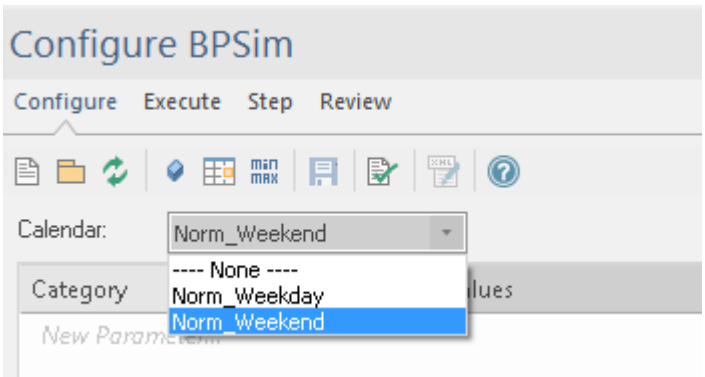
You can add a new calendar, or edit or delete a selected calendar.

2. To add a new calendar period, click on the New button to display the 'Event Recurrence' dialog.



3. In the 'Event time' panel, the 'Start' and 'End' fields both default to the current time. The 'Start' field is the anchor; a change to either the 'End' field or the 'Duration' field automatically updates the other field, in reference to the 'Start' field. Click on the hour and minute segments of each field (and, for the 'Duration' field, the 'Day(s)' segment) separately, and use the 'spin' arrows to set the start time and the end time or duration of the period.
4. In the 'Recurrence pattern' panel, select the radio button for the interval at which the calendar period recurs. Each option displays an appropriate set of fields in the right of the panel for refining that interval to every day/week/month or every two/three/four days/weeks/months, on a particular day of the week, or day or date of the month, or day or date in the year. Select the checkboxes or values in drop-down lists as appropriate.
5. In the 'Range of recurrence' panel, select the date on which the calendar period takes effect and select the appropriate radio button to define when the period ceases to apply - never, after a set number of occurrences, or on a specific date. You can select an end date either from a drop-down calendar or using the 'spin' arrows on each segment of the date.
6. Click on the OK button to set the calendar period.

As you define calendar periods, they are listed in order of the start date and/or time, earliest first.



With defined calendars, you can configure parameters on a selected calendar.

Validation

After configuring BPSim parameters for some BPMN elements, click on the button to run a validation of the simulation. Any BPMN or BPSim errors/warnings will be displayed in the System Output window. Fix the issues according to the messages.

After doing this, proceed to the next Help topic: *BPSim Execute Page*.

BPSim - Execute Page

After defining the configuration, you can choose to conduct a Standard simulation or a Customized simulation. The execution will generate a result report and a list of records used to replay (step through) the simulation.

Access

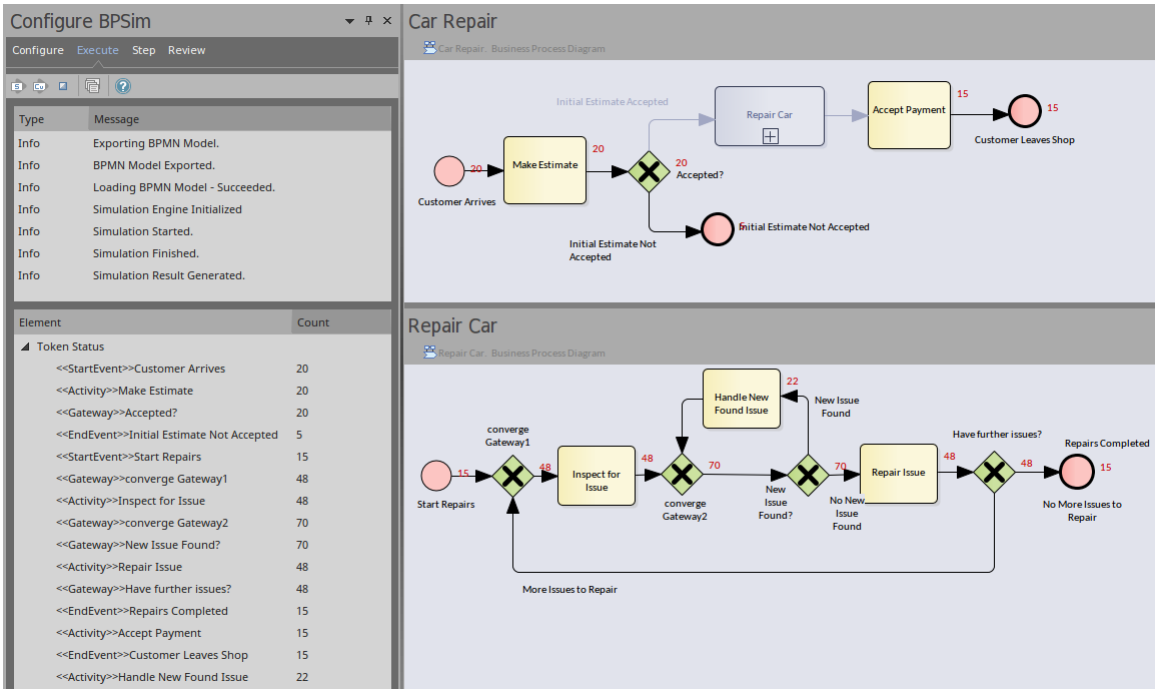
Ribbon	Simulate > Process Analysis > BPSIM > Open BPSIM Manager > Execute page

Toolbar Options

Option	Description
	Click on this button to execute the BPMN model with the BPSim configuration and generate a standard report.
	Click on this button to execute the BPMN model with the BPSim configuration and generate a customized report based on the 'Result Request' settings made on the 'Configure' page.
	Click on this button to stop the execution and exit the simulation.
	Click on this button open the generated report in the 'Review' page.

Execution

When you click on the Run Simulation button or Run Customized Simulation button, the BPMN model with BPSim Configuration will be exported and loaded into the Execution engine.



During the simulation:

- The token status list will flash with runtime values
- The diagram will flash with runtime token counts

However, the simulation might execute too quickly to be able to see this. You can see these changes if you use the 'Step' page to run through the simulation step by step.

In this example, BPMN elements under the process 'Car Repair' and sub-process 'Repair Car' get triggered as new customers arrive at regular intervals.

BPSim - Step Page

After successfully running the execution, the system generates an execution report that tells you the process status in general, such as (for the Car Repair example) the average time of a task, the total waiting time of customers and how many issues are repaired.

In addition, you can inspect the process from various angles. For example:

- From the timestamp - what was the status of this process at 9:30 AM?
- From the token - what did the 3rd customer do in the shop?
- From the property - how does the number of issues decrease and increase for the 2nd car?
- From multiple threads - can I see customers walk in and simulate automatically on the diagram?
- From the resources - when is a support person busy or idle? Why is a customer waiting for 40 minutes?

All of these kinds of question can be answered on the 'Step' page.

Access

Ribbon	Simulate > Process Analysis > BPSIM > Open BPSIM Manager > Step page

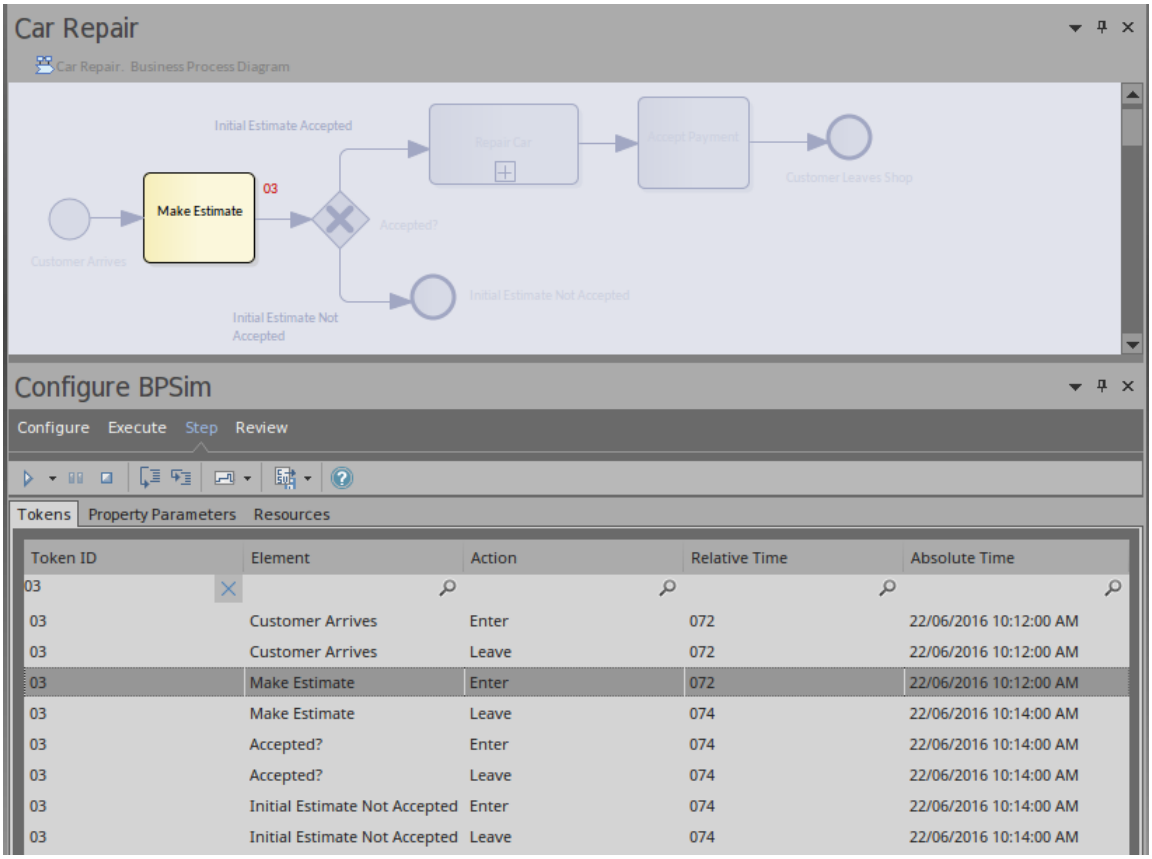
Toolbar Options

Option	Description
	Click on this button to simulate the process automatically based on the execution result. Click on the drop-down arrow and on the menu option 'Set Speed for Replay', and adjust the simulation speed as a multiple of normal. For example, typing '60' make the simulation 60 times faster than the actual activity; 1 minute in real life will be simulated in 1 second.
	Click on this button to pause the auto-replay simulation.
	Click on this button to stop the simulation.
	Click on this button to 'Step over' to the next timestamp. Each 'Step over' can contain multiple 'steps'.
	Click on this button to play a single step. This represents a single movement of a token in the process.
	Click on this button to generate a Timing Diagram for the simulation. You can choose from the menu, either 'Generate a single timeline for each token' or 'Generate multiple timelines for each token'. See *Generate Timing Diagram* later in this topic.
	Click on this button to export the filtered records on this step page to a CSV file.

	You can choose from which tab ('Tokens', 'Property Parameters' or 'Resources') the data is exported.

Tokens Tab

After running the execution, this page will be filled with token information during the simulation; the sequence of entries is in order of triggering time.

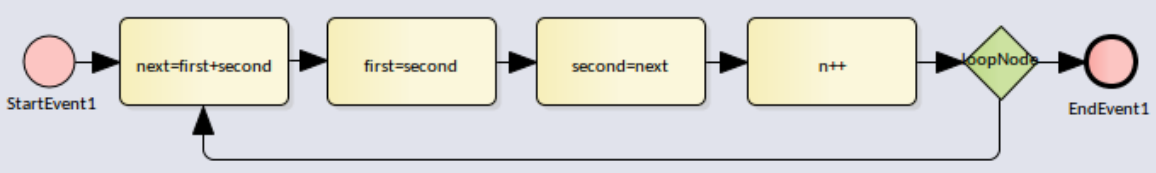


- Using the Filter Bar in the Header band (right-click on the column heading and select 'Toggle Filter Bar') you can filter the results shown; for example, typing 03 in the 'Token ID' column will show only the records for token 03
- If you click once on the Step in button, one record in the list will play
- If you double-click on a record the simulation will 'Step to' that record from the beginning
- If time parameters are set on the elements, clicking on the Step over button will run to the last record of the next time event
- When a record in the list is played, the simulation snapshot will show on the diagram

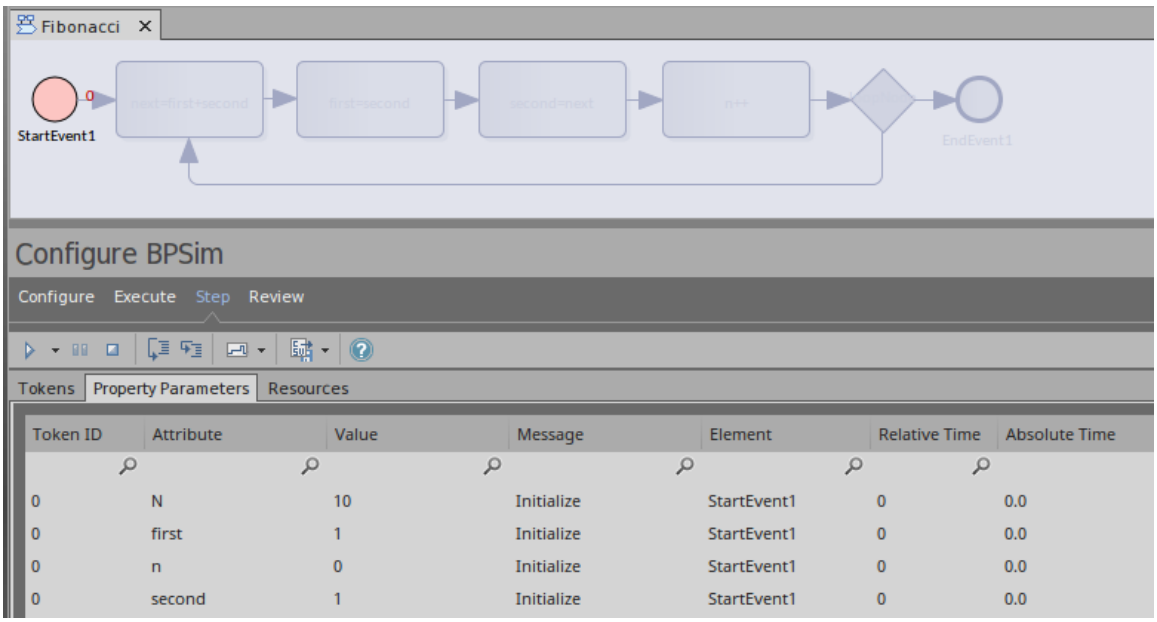
Property Parameters Tab

While the records on the 'Tokens' tab are played, the 'Property Parameters' tab will show the runtime value of properties at the timestamp.

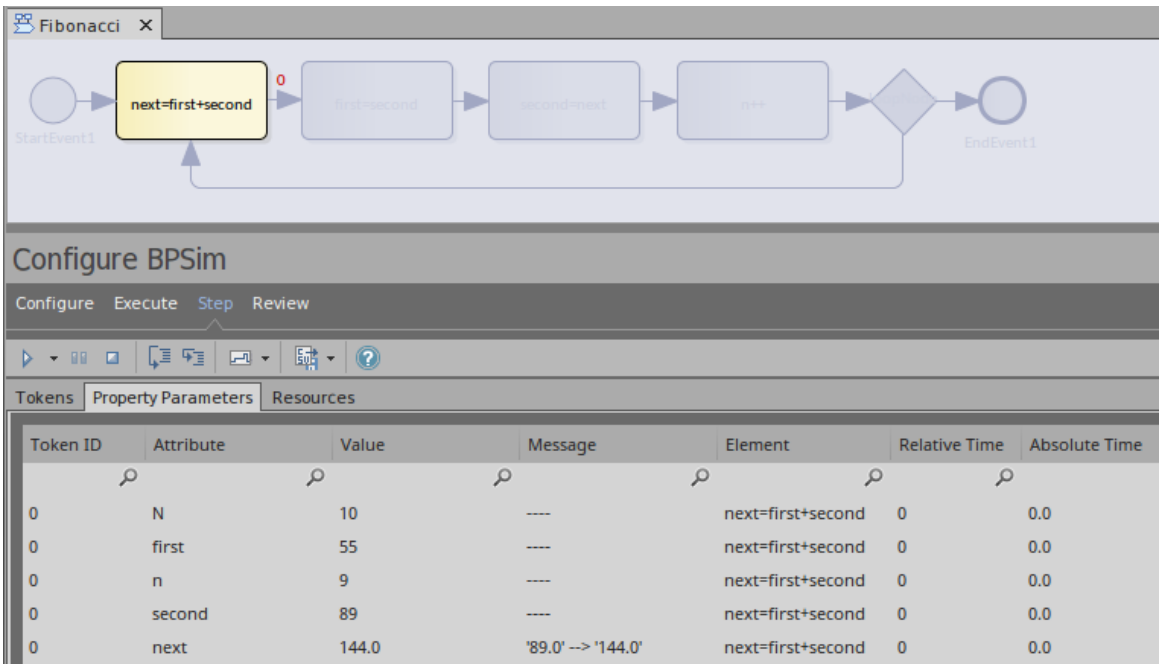
For example, a BPMN process to calculate Fibonacci numbers might be modeled in this way:



After defining property parameters, configuring BPSim parameters for each element and executing the model, we are ready for the step simulation:



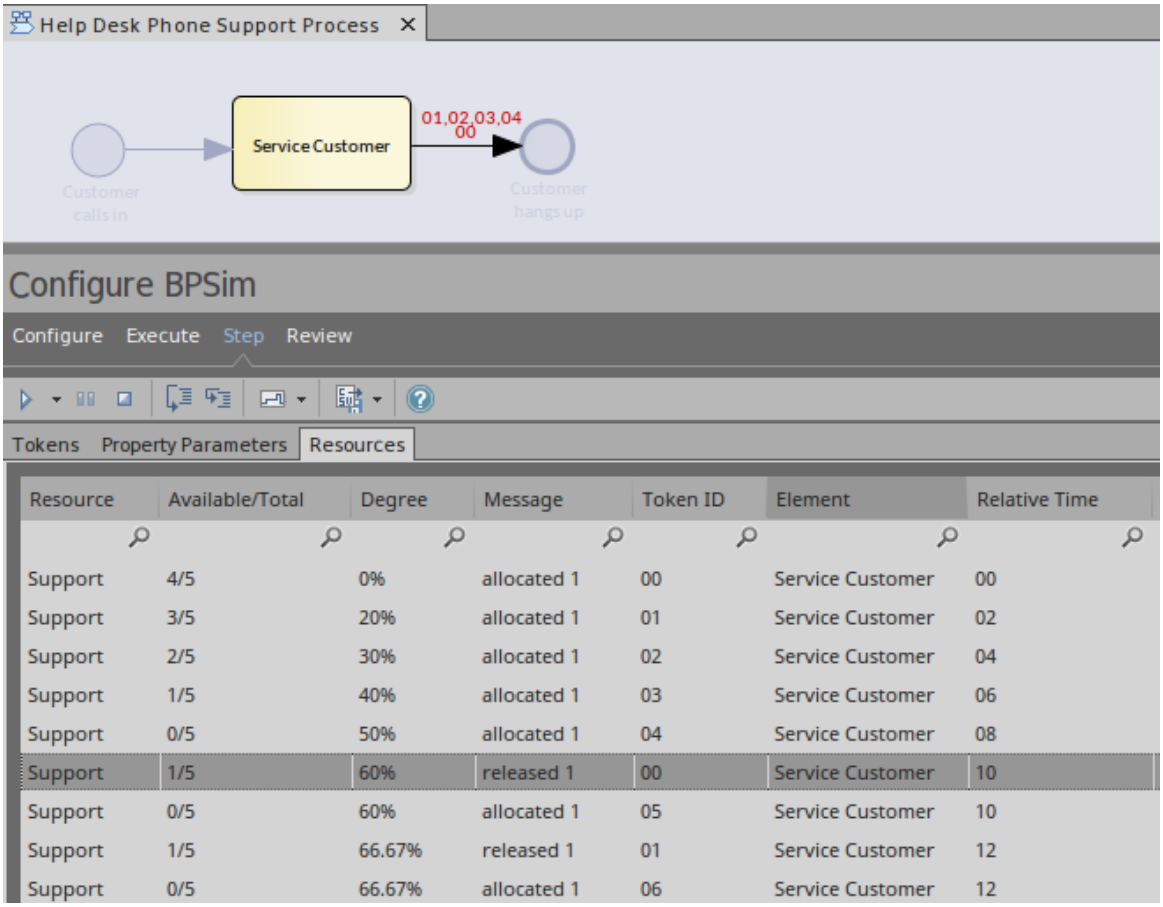
The 'Message' column indicates that properties 'N', 'first', 'n' and 'second' are initialized.



If you keep clicking on the Step in button, the properties in the list will change their values. The illustration shows that on entering task 'next = first + second', the value of property 'next' changes from 89 to 144.

Resources Tab

While the records on the 'Tokens' tab are played, the 'Resources' tab will show the runtime resource available, the quantity of that resource available and the allocation or release events at the timestamp.



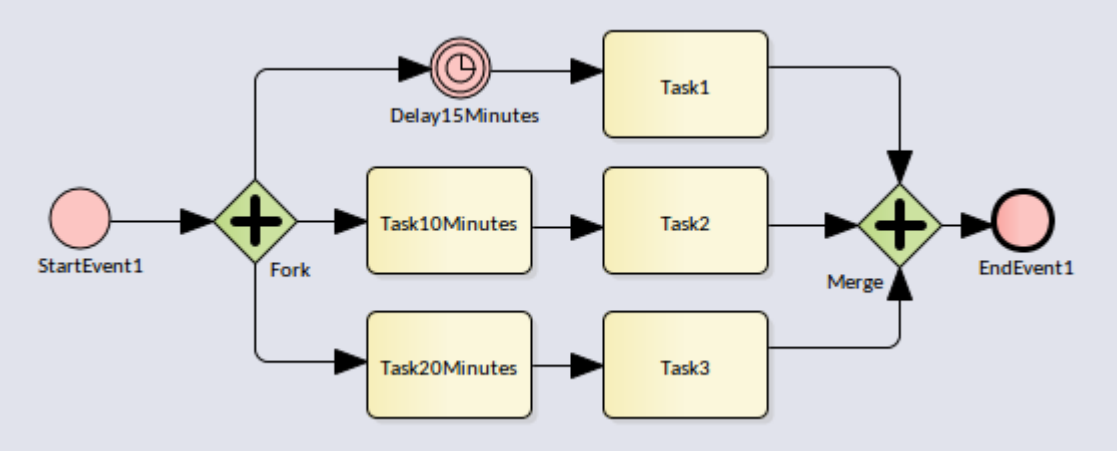
Resource	Available/Total	Degree	Message	Token ID	Element	Relative Time
Support	4/5	0%	allocated 1	00	Service Customer	00
Support	3/5	20%	allocated 1	01	Service Customer	02
Support	2/5	30%	allocated 1	02	Service Customer	04
Support	1/5	40%	allocated 1	03	Service Customer	06
Support	0/5	50%	allocated 1	04	Service Customer	08
Support	1/5	60%	released 1	00	Service Customer	10
Support	0/5	60%	allocated 1	05	Service Customer	10
Support	1/5	66.67%	released 1	01	Service Customer	12
Support	0/5	66.67%	allocated 1	06	Service Customer	12

Generate Timing Diagram

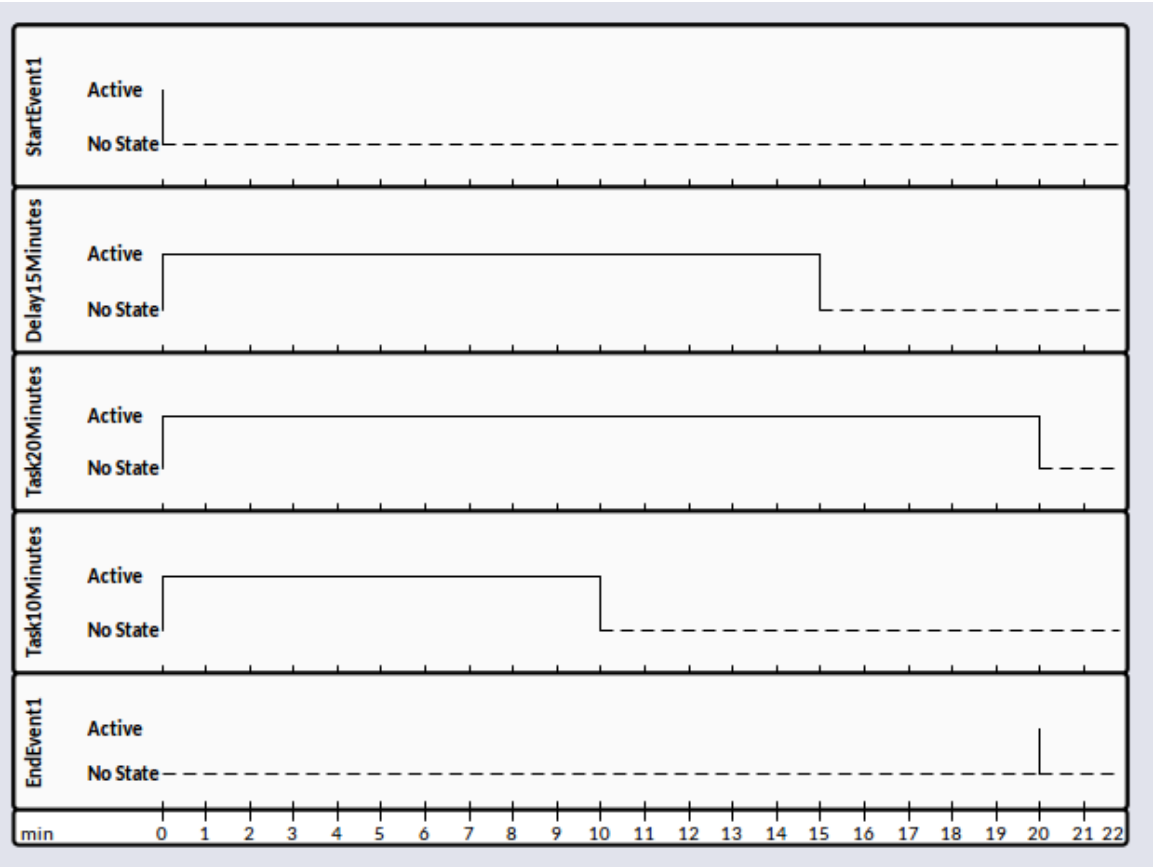
When time parameters are configured on the BPMN elements, Enterprise Architect can generate a Timing diagram for the simulation process.

- Generate a single timeline for each token - use this option for a 'single threaded' process; that is, no Parallel Gateway or Event sub-processes
- Generate multiple timelines for each token - use this option in cases where the 'Generate a single timeline for each token' option does not apply

For example:



Execute this model and click on 'Generate multiple timelines for each token' the generated Timing diagram resembles this:



BPSim - Review Page

This review page contains three tabs:

- Configuration Summary
- Standard Results Report
- Custom Results Report

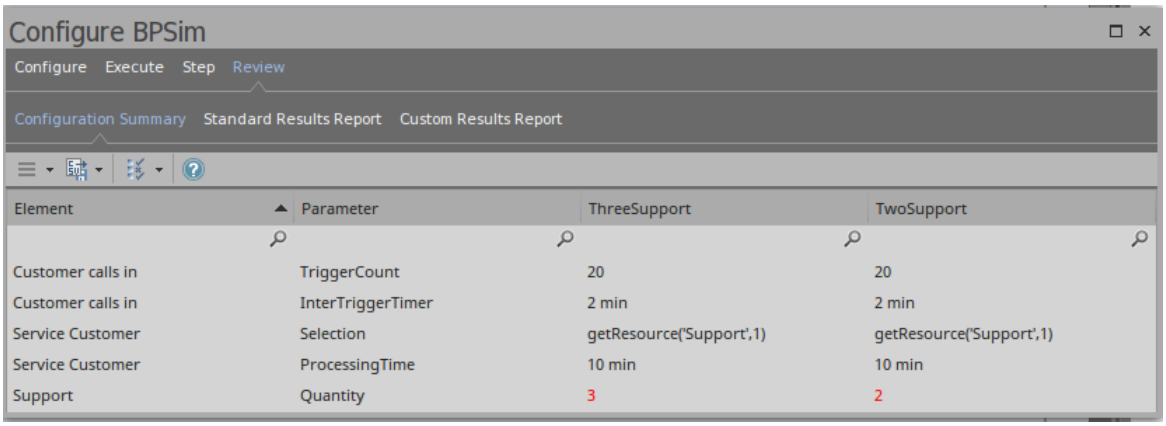
These tabs work in a similar way: add one Artifact for review or multiple Artifacts for comparison. This makes it easy for you to do what-if analyses.

Access

Ribbon	Simulate > Process Analysis > BPSIM > Open BPSIM Manager > Review

What-If Analysis

In the Help Desk Support example, we can compare two Artifacts and their corresponding results.



In this illustration, we have clicked on the icon in the toolbar and selected the 'Show only different items' option to see what differences the changed parameter values have caused.

Element	Parameter	ThreeSupport- Result	TwoSupport- Result
Help Desk Phone Support Process	Maximum Time	34	64
Help Desk Phone Support Process	Standard Deviation Time	7.72	17.23
Help Desk Phone Support Process	Average Time	21.4	37
Service Customer	Average Number Of Tokens Waiting For Resource	3.17	5.29
Service Customer	Average Time In Task	21.4	37
Service Customer	Average Time Waiting For Resource	11.4	27
Service Customer	Maximum Number Of Tokens Waiting For Resource	8	12
Service Customer	Maximum Time In Task	34	64
Service Customer	Maximum Time Waiting For Resource	24	54
Service Customer	Total Time In Task	428	740
Service Customer	Total Time Waiting For Resource	228	540
Support	Degree Of Utilisation	92.59%	98.04%
Support	Number Started Immediately	3	2
Support	Total Time Available	216	204
Support	Total Time Idle	16	4
Support	Average Number Available	0.22	0.04

We see that when the number of support staff decreases from 3 to 2, the average time waiting for resources increases from 11.4 minutes to 27 minutes.

Using the Parameter Value Dialog

The 'Parameter Value' dialog helps you to define values for a wide range of parameters throughout the BPSim Configuration. It supports the definition of simple fixed values through to distributions and expressions that yield a derived value. Not all types of value or derivation are appropriate to all types of parameter.

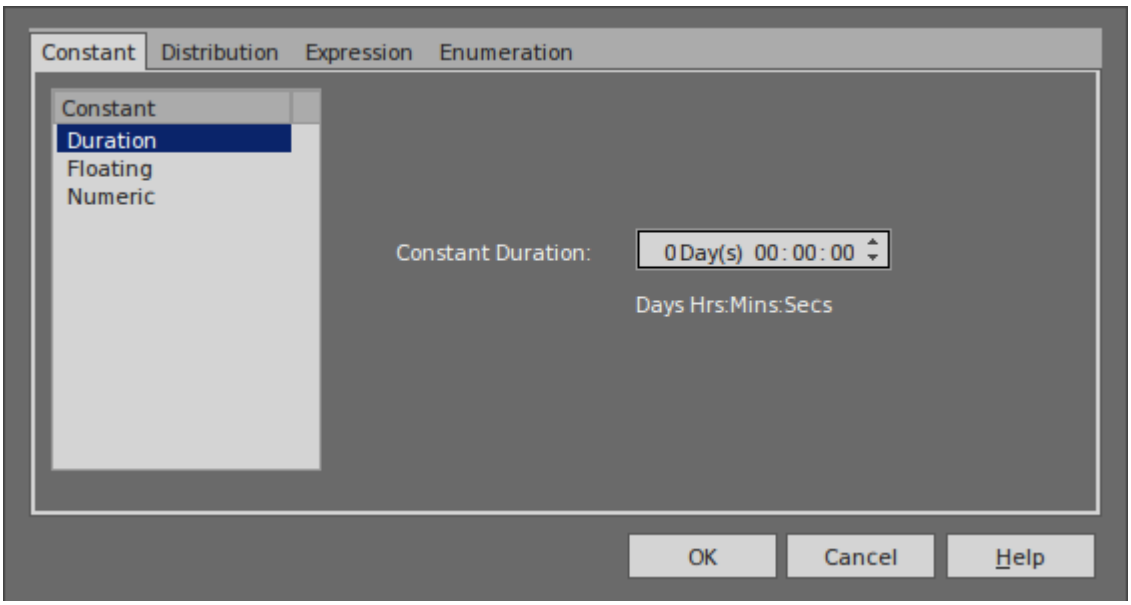
The dialog name is taken from the object name and the name of the parameter being defined; for example, Configure 'Processing' for 'Activity1'.

Access

With a BPSim Artifact loaded in the Configure BPSim window, select a BPMN Element on the diagram or Browser window, then click on ... in the 'Values' field. (If the parameter is not already created, choose Category and Parameter in the list to create a new one.)

Constant Tab

Use this tab to define a specific value for the parameter - a numeral, text string or time, for example.



In the 'Constant' panel, select the type of constant:

- Floating
- Numeric
- String
- DateTime
- Boolean, or
- Duration

Appropriate fields display to the right of the panel; type the value and, if required, the unit in which the value is expressed (for example, a unit of time or of currency). For some types of parameter a drop-down list is available from which you can select a value.

Distribution Tab

On this tab, you can apply a statistical sampling method to obtain the parameter value; for each type of distribution available, the appropriate fields display for you to enter the parameters of the distribution. All distributions require you to identify the unit of expression.

The distribution parameters are not necessary for the business process you are developing, but (if you are deriving values from a distribution) are required for the simulation.

You can select from these types of distribution:

- **Beta** - a continuous probability distribution providing 'real' values within a short range, commonly 0 to 1
- **Weibull** - a continuous probability distribution providing 'real' values, commonly used for object lifetime analysis
- **Gamma** - a continuous probability distribution providing 'real' values, useful for modeling exponentially distributed random variables
- **Binomial** - an 'integer' distribution, providing values based on the number of trials and the probability of a certain outcome
- **Erlang** - provides 'real' values based on the Erlang *K* value and the mean of the distribution
- **Normal** - provides 'real' values based on the mean and standard deviation of the distribution
- **LogNormal** - a continuous probability distribution of 'real', random variables whose logarithm is normally distributed
- **Poisson** - a discrete ('integer') probability distribution that expresses the probability of a given number of events occurring independently in a fixed interval of time or space (volume, distance or area)
- **NegativeExponential** - provides 'real' values based on the mean of the distribution
- **Triangular** - provides 'real' values based on the mode of the distribution and the minimum and maximum values of a range
- **TruncatedNormal** - provides 'real' values based on the mean and standard deviation of points within the minimum and maximum values of a range
- **Uniform** - provides 'real' values between the minimum and maximum values in a range

Expression Tab

On this tab, you type in an XPATH 1.0 expression to combine explicit values, operators and functions to be processed at runtime to provide a value. Each property parameter of an expression must be enclosed in braces - {xxx}.

Example 1: In order to represent c = a + b + 10, we assign this expression to a property 'c':

{a} + {b} + 10

where 'a' and 'b' are properties defined in the BPSim model.

Example 2: In order to represent $c = t - p * (a - b)^2$, we assign this expression to a property 'c':

{t} - {p} * Math.pow({a} - {b}, 2.0)

Note: When simulating a model with this expression, please select 'Java' as the language in order to use the java built-in function Math.pow().

Enumeration Tab

On the 'Enumeration' tab, you can define an enumeration to supply a collection of constant values. You would have obtained these values from real-world, historical data or from analysis and simulation of a model. Every time the

parameter is evaluated, the next enumeration value is returned.

As you define each numeration value, click on the Save button to add it to the list of possible values, and click on the New button to clear the data fields ready to enter another value. For some types of enumeration value you might be asked to define the unit in which the value is expressed. The types of enumeration you can define include:

- String
- Floating
- Numeric
- Duration
- DateTime
- Boolean

Using the BPSim Execution Engine

The BPSim Execution Engine is an Add-In integrated with the Unified and Ultimate editions of Enterprise Architect, to execute the simulations that you have defined using the Business Process Simulation (BPSim) facility. The Engine is a prerequisite for accessing and using the facilities of BPSim.

Access

Open the Configure BPSim window using one of the methods in this table, click on the button and browse for a Business Process Simulation Artifact.

Ribbon	Simulate > Process Analysis > BPSim > Open BPSim Manager (or Simulate > Process Analysis > BPSim > Find BPSim Configuration Artifacts)
Context Menu	Right-click on a Business Process Simulation Artifact element | Configure BPSim
Other	Right-click on a Business Process Simulation Artifact element | Simulate BPMN with BPSim

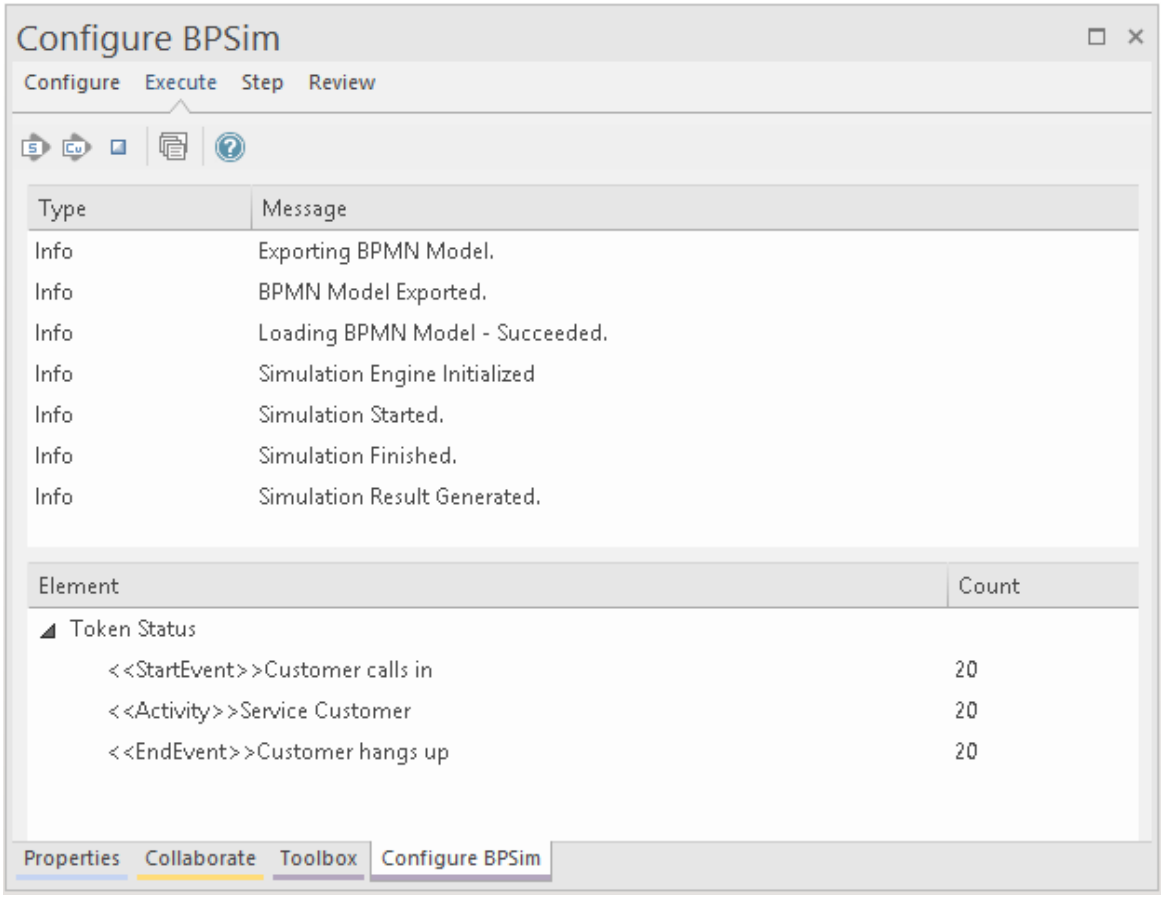
Execute and Control a Simulation

Click on the 'Execute' tab and on either:

- to start a Standard Simulation or
- to start a Customized Simulation

These options trigger the same processing, except that while a Standard Simulation generates a report on all built-in parameters set in the simulation, a Customized Simulation extracts results for only the parameters you have specifically flagged using the 'Result Request' columns in the configuration.

The simulation executes, displaying processing messages in the top section of the dialog, and the elements and parameters processed with the runtime values used from the configuration.



While the simulation is in progress, you can click on the icon to cancel the simulation.

The results of the simulation are written to an Artifact element added to the Business Process parent Package. A Standard simulation writes to a <<BPSimReport>>-stereotyped Artifact, whilst a Customized simulation writes to a <<BPSimCustomReport>>-stereotyped Artifact.

Track Property Values

As well as the built-in parameters, you can define your own process-specific Property parameters (attributes) in the configuration. When the simulation has finished, and if you have defined property parameters, the Attributes button is enabled. When you click on this button the 'BPSim PropertyParameter Values' dialog displays, through which you can track how the run-time values of all the property parameters accrue or change through the business process.

Review a Simulation

When the simulation has finished processing, click on the Open Result button. The 'BPMN Simulation Report View' tab opens in the main work area, showing the results for the built-in parameters in the current simulation (but not for the user-defined property parameters). If you have already run a simulation of another configuration based on the same business process, that is also displayed in the report as an additional column. Otherwise, you can click on the report Artifact element and drag it onto the report tab, to compare the runtime values of the built-in parameters under two (or more) configurations.

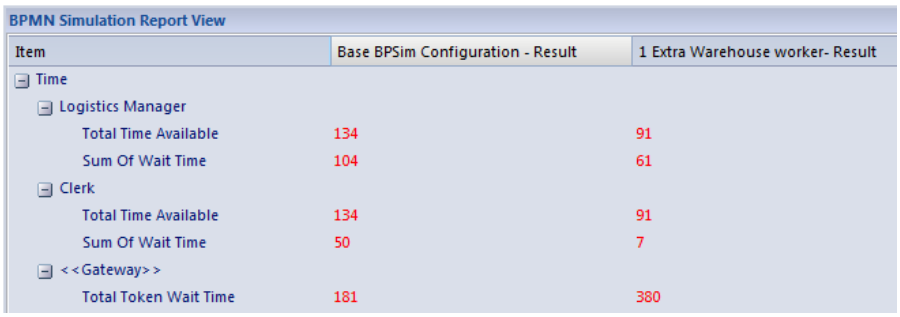
To make it easier to view the data from the report, you can drag the 'BPMN Simulation Report View' tab out of the main view so that it becomes a floating window, and enlarge the window to a suitable size.

Click on the expansion boxes against the parameters you want to check. You can also expose and filter the information using right-click context menu options.

You can represent specific differences between the results from separate simulations as Charts. The simulation result Artifacts (<<BPSimReport>>*name* - Result elements) must exist before you can set up the Chart Artifacts. There is a template Chart Artifact for Standard simulations and one for Customized simulations.

BPMN Simulation Report Options

Option	Description

Collapse All	Select this option to collapse the parameter hierarchy down to just the parent tab names.
Expand All	Select this option to expand the parameter hierarchy out to the lowest value type.
Show Only Different items	(When you have two or more simulations shown.) Select this option to restrict the display to those parameters where the values differ between the simulations. Click on the option again to deselect it.
Highlight Different Items	(When you have two or more simulations shown and where some of their parameter values are different.) Displays the differing parameter values in red. This option is disabled if you select the 'Show Only Different Items' option. 
Show Only Non Empty Items	Select this option to filter the display to show only parameters that have a specific value other than 0.
Remove Model	(When you have selected a specific result, which identifies the simulation in the report.) Select this option to remove the simulation column from the report.

BPMN Simulation Report View

Item	Base BPSim Configuration - Result	1 Extra Warehouse worker- Result
Time		
Logistics Manager		
Total Time Available	134	91
Sum Of Wait Time	104	61
Clerk		
Total Time Available	134	91
Sum Of Wait Time	50	7
<<Gateway>>		
Total Token Wait Time	181	380

BPSim Execution Engine - Simulation Language

The BPSim Execution Engine supports simulation on XPath 1.0 or Java, where the appropriate language is defined as the Expression Language in the simulation configuration. It also supports the use of process instance data in BPSim Property Parameters, where the actual value is only determined during execution.

XPath 1.0 Operators

These operators can be used in BPSim expression parameters.

Operator	Description
|	The Union operator, used for resource acquisition. Example: getResource('w1',1) | getResource('w2',1)
+	Addition. Example: 4 + 6
-	Subtraction. Example: 6 - 4
*	Multiplication. Example: 6 * 4
div	Division. Example: 8 div 4
=	Equality. Example: 4 = 4 (True)
!=	Not Equal. Example: 5 != 3
<	Less than. Example: 6 < 9
<=	Less than or equal to. Example: x <= 6
>	Greater than. Example: 9 > 6
>=	Greater than or equal to. Example: n >= 7
or	Alternative. Example: n = 6 or n <= 6

and	Combination. Example: n = 5 and m < 8
mod	Modulus division. Example: 5 mod 2
getProperty	Get a property value. Example: getProperty ("amount")
getResource	Get a resource assignment. Example: getResource ('w1',1)

Note

The Expression Language can be set in the Configure BPSim window, on the 'Configure' tab; the two options 'XPath 1.0' and 'Java' are available as values of the 'Expression' parameter.

If you select 'Java', you must set the property 'JDK Home' to a valid JDK directory.

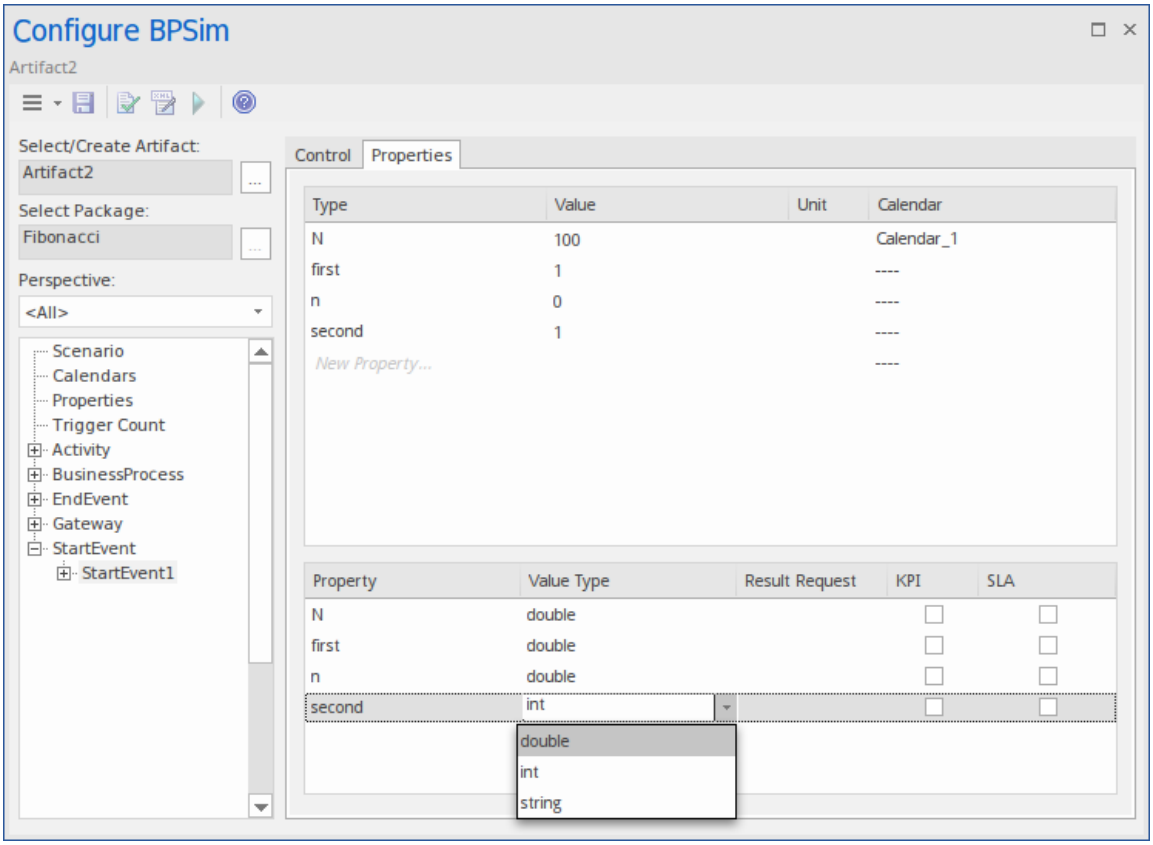
Tip: You can use {PropertyParameterName} as a short form of getProperty('PropertyParameterName'), which is useful when writing the value for the expressions; for example:

{n} < {N} instead of getProperty('n') < getProperty('N')

The short form of the getProperty operator can be used in both XPath 1.0 and Java.

BPSim Property Parameters

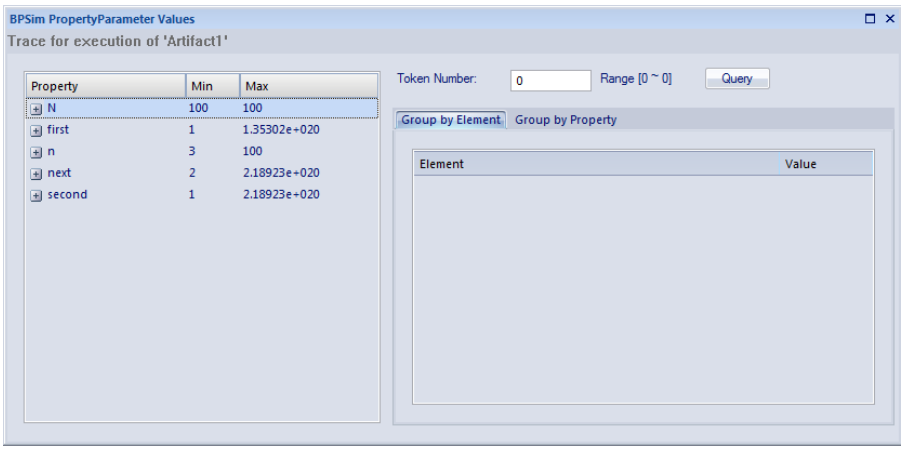
From Enterprise Architect release 13.0 onwards, BPSim property parameters can hold process instance data to which no value is assigned until run time. You can define the property parameter type on the 'Properties' page of the Configure BPSim window; the supported types are 'int', 'double' and 'string'.


Configure BPSim
Artifact2
Select/Create Artifact:
Artifact2
Select Package:
Fibonacci
Perspective:
<All>
Scenario
Calendars
Properties
Trigger Count
Activity
BusinessProcess
EndEvent
Gateway
StartEvent
StartEvent1
Control
Properties
Type
Value
Unit
Calendar
N
100
Calendar_1
first
1
n
0
second
1
New Property...
Property
Value Type
Result Request
KPI
SLA
N
double
first
double
n
double
second
int
double
int
string

Tracking Property Parameter Values

The Business Process Simulator (BPSim) helps you to model and test the operating details of a business process, such as the resourcing of activities and tasks, the intervention of events, and the impact of decision points and the decisions made at those points. You add these process-specific property parameters, or attributes, to the BPSim configuration and, as you run simulations on the model according to the configuration, the BPSim engine helps you to capture the run-time values of the property parameters for every iteration of the simulation, and to filter the results to examine specific pathways or decision points. This gives you an incredibly detailed insight into what might actually happen in your business process under a specific condition or combination of conditions, either to generate a result or to show what processing path produces that result.

Access

Context Menu	Right-click on a defined Business Process Simulation configuration Artifact |Simulate BPMN with BPSim... : Run (select simulation type) : Attributes (The Attributes button is not available if the simulation configuration does not contain any property parameters) 

BPSim PropertyParameter Values dialog fields

Option	Action
Property	This table lists the properties defined for the process, and shows the minimum and maximum possible values for each property for the whole process. If you click on the expansion box for a property, the table shows the minimum and maximum values for the property at each activity or event (element) during the process.
Token Number	Type the number of the 'token' to examine; this number must lie within the range shown to the right of the field. A 'token' is an independent trigger, such as a customer or an order entering a business and initiating the business process under review. There can be any number of customers or orders each having no relationship with any other customer or order, and each potentially following a different route through the business process. Where there is only one possible instance of one possible trigger event, such as an

	on/off switch being thrown, the token is regarded as being 0.
Query	Click on this button to initiate a query on the token simulation, to populate the 'Group by Element' and 'Group by Property' tabs.
Group by Element	Displays the results from the perspective of how a property parameter's value changes within a selected element. The tab shows a list of the elements in the process and, for each element, the value of each property applied in the element on each iteration of the simulation. Using the 'Toggle Filter Bar' option on the header bar, you can refine the display to show only a particular property and see how often it is used by the element and with what values.
Group by Property	Displays the results from the perspective of how the value of each property changes during the whole process. The tab shows a list of properties that are applied during the process and, for each property, the value in each activity (element) on each iteration of the process.

Examples

In the EAExample Model, you can study two examples of generating information on property parameters from a simulation of a BPMN business process model. These will demonstrate how you define property parameters in the configuration, based on the model. You can initially just run a simulation on each example, and examine the output as described here. You can then examine the business processes and configurations themselves, and change or add to the property parameters provided.

The examples are described in the *Tracking Property ParameterValues - Examples* topic. Briefly, they are:

- 'Fibonacci' - a very simple recursive business process that calculates a series of Fibonacci numbers through ten iterations; you can see how the property parameters increment in each iteration through the elements of the process (in Example Model > Model Simulation > BPSim Models > Fibonacci)
- 'Car Repair' - a more complex and realistic process that represents what might happen when a series of individual 'walk-in' customers bring vehicles into a car repair shop for estimation and repair (in Example Model > Model Simulation > BPSim Models > Car Repair Process)

There is also a small example of the behavior of the Time Parameters (in Example Model > Model Simulation > BPSim Models > Time Parameter).

Notes

- If a BPSim configuration contains result requests and a Custom simulation is performed on it, the 'BPMN Simulation Report' shows only the built-in parameters requested in the configuration; in contrast, the 'BPSim PropertyParameter Values' dialog lists all property parameters regardless of any result request settings or type of simulation

Tracking Property Parameter Values - Examples

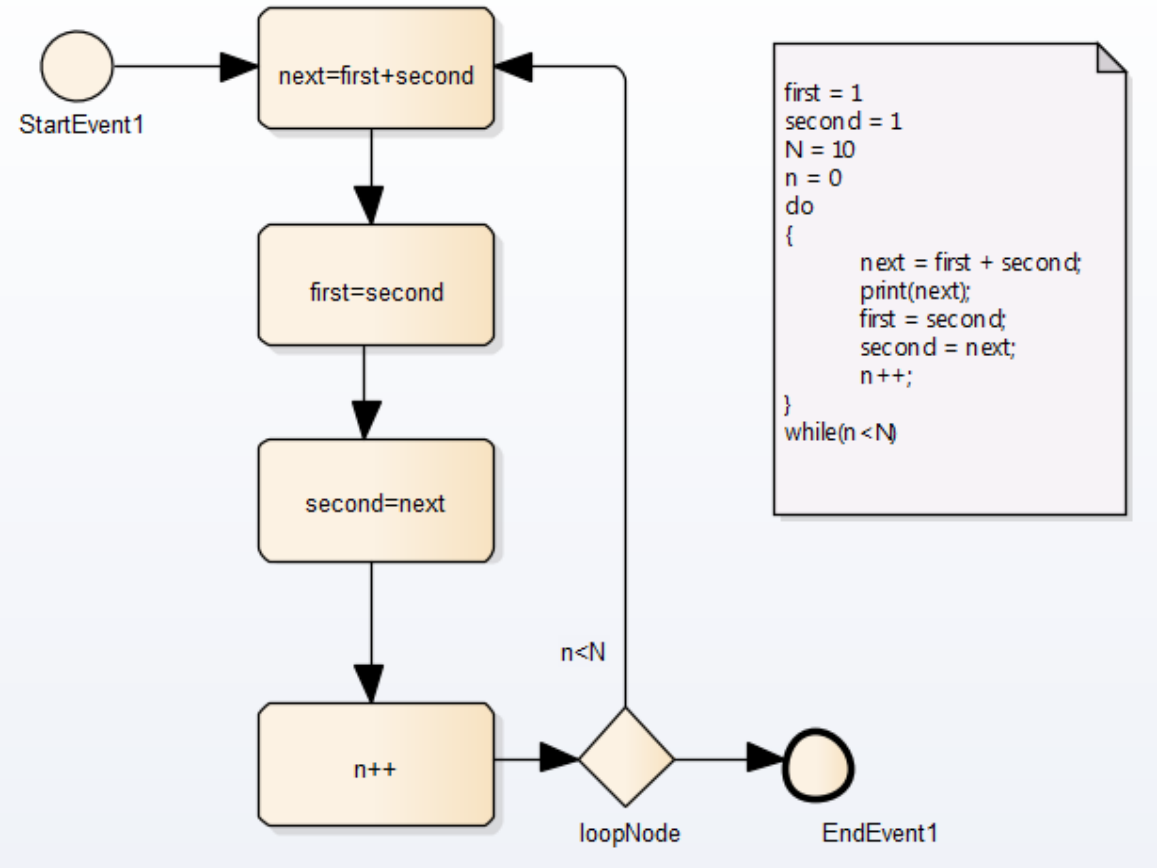
To help you learn about the facility for generating information on property parameters from a simulation of a BPMN business process model, Sparx Systems provide two examples that you can explore in the EAExample model. These are the:

- Fibonacci process - a very simple example to help you become familiar with the parameter tracking facilities
- Car Repair process - a more complex example that you can manipulate to see how a real-life process might be investigated

At the end of this topic is a section that briefly discusses how you might work with an integer-based process containing parameters that are initialized by 'real' distributions, and a section describing the example of the behavior of the Time Parameters.

The Fibonacci Example

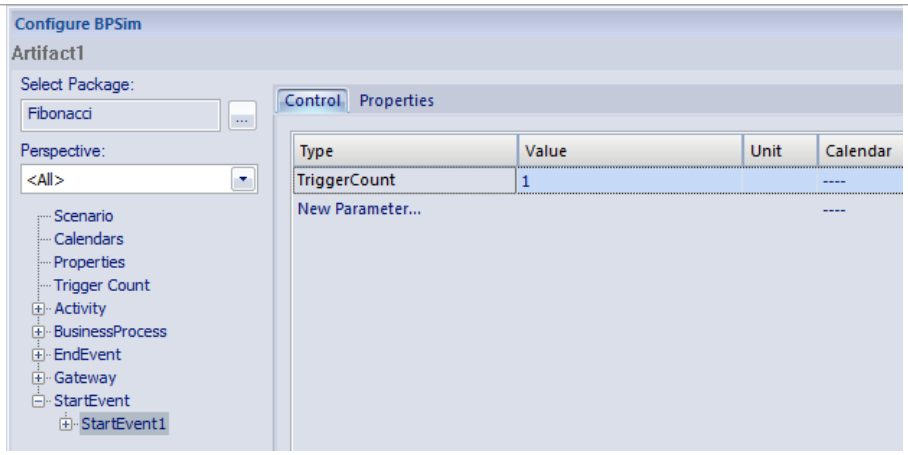
This is a very simple recursive business process that calculates a series of Fibonacci numbers through ten iterations; you can see how the property parameters increment in each iteration through the elements of the process. Open Example Model > Model Simulation > BPSim Models > Fibonacci.



The pseudocode of the process is shown in the Notes element on the diagram. The statement 'print(next)' will output the number series 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

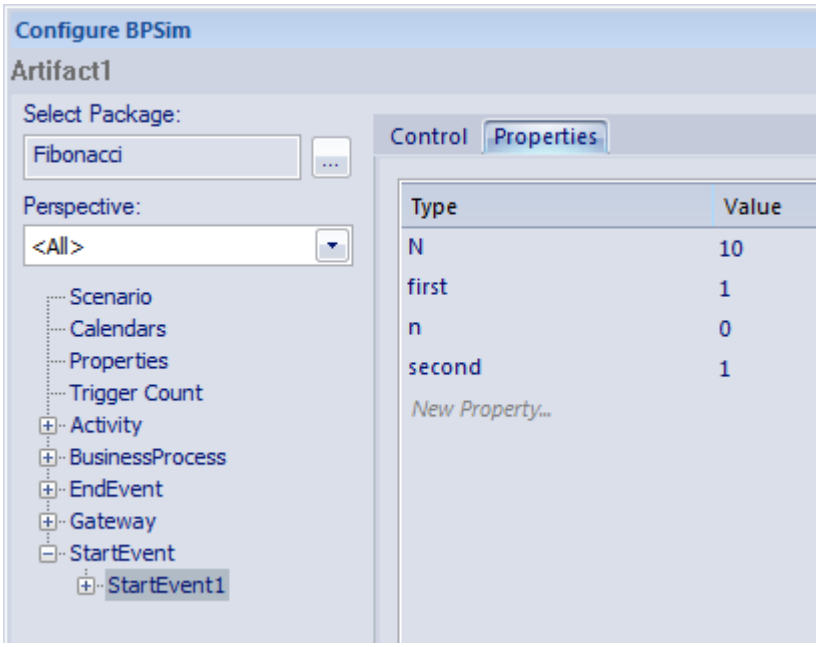
The BPSim configuration for this process is set up as described here.

Step	Action
1	



On the 'Control' tab for the StartEvent element, set 'TriggerCount' to '1', and on the 'Properties' tab create and initialize the properties:

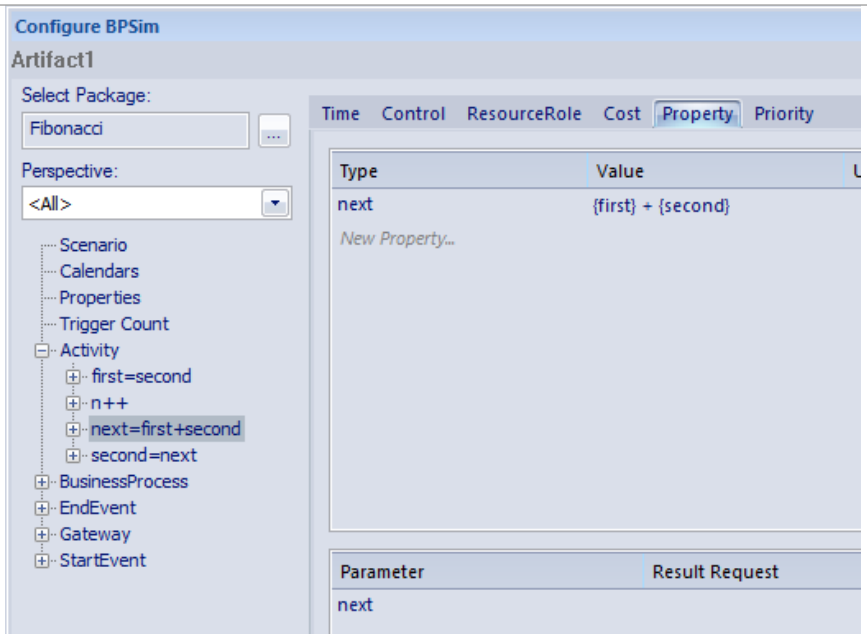
- 'N' as '10'
- 'first' as '1'
- 'second' as '1'
- 'n' as '0'



2

Now define the properties for each of the Activities in the process, on the 'Properties' tab. Note that the values for these properties are derived from **Expressions**, the components of which must be enclosed in braces - {xxx}. For the Activity:

- next=first+second - set the property 'next' and define the value as the Expression {first} + {second}



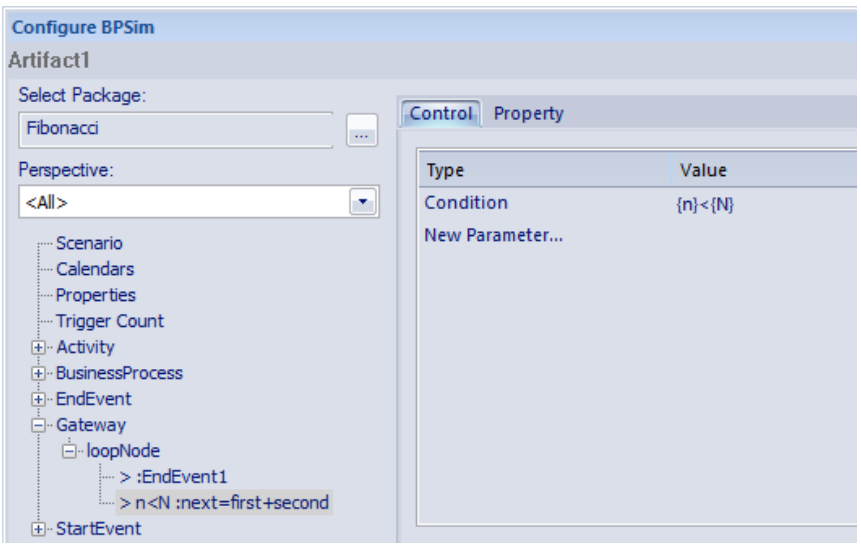
- first=second - set the property 'first' and define the value as the Expression {second}
- second=next - set the property 'second' and define the value as the Expression {next}
- n++ - set the property 'n' and define the value as the Expression {n} + 1

3

Set the 'Condition' property parameters for the two Sequenceflow connectors issuing from the 'loopNode' Gateway element, on the 'Control' tab.

Expand the Gateway | loopNode element and for the link to:

- next=first+second - set the Control parameter to 'Condition' and define the value as the Expression {n}<{N}

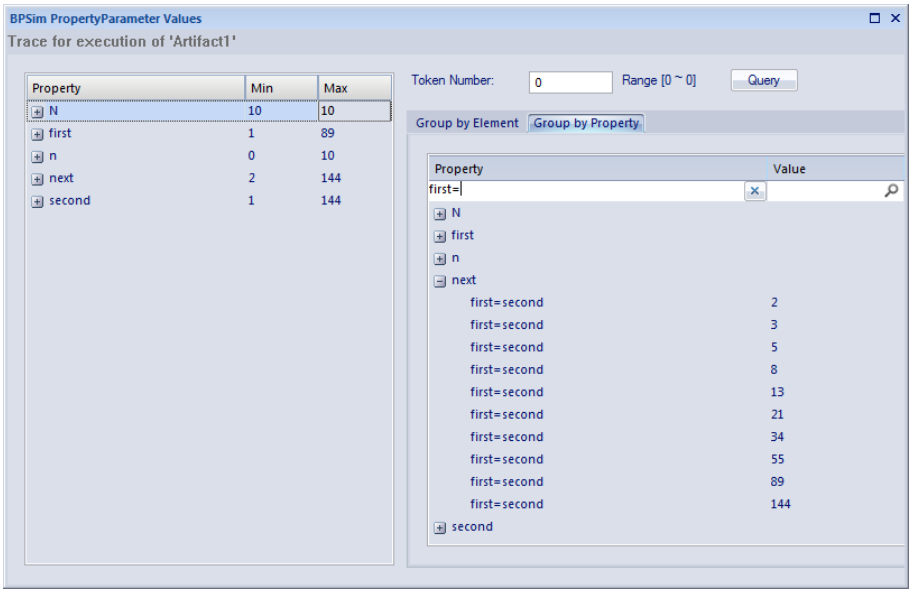


- EndEvent1 - set the Control parameter to 'Condition' and define the value as the Expression {n}=>{N}

4

Having completed the configuration, click on the Run button on the Configure BPSim window and on the 'BPSim Simulation Controller' dialog, selecting a Standard simulation.

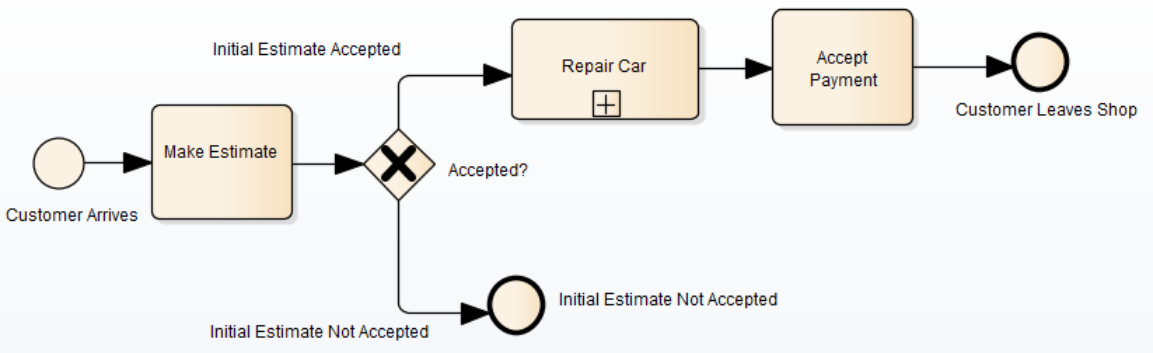
When the simulation is complete, click on the Attributes button.

	On the 'BPSim PropertyParameter Values' dialog, set the 'Token number' field to '0' and click on the Query button.
5	Now examine the values of the 'next' property on entering the first=second Activity in each iteration of the simulation. Click on the 'Group by Property' tab and expand the 'next' item. The list of values is long, so right-click on the column headers and select the 'Toggle Filter Bar' option. Under the 'Property' column heading, type 'first='. This filters the list to show only the property parameter values on entering the first=second Activity. 

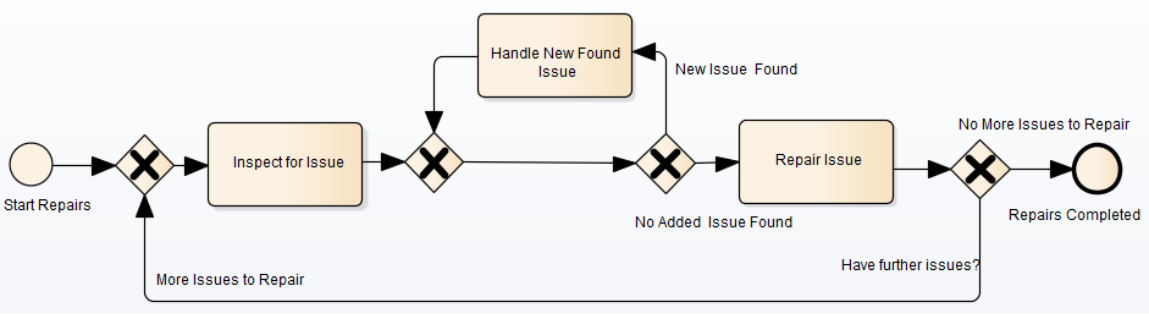
The Car Repair Example

This more complex example is based on a realistic model of a car repair process, where a number of individual customers request an estimate for repair and either proceed with the repair or decline to continue; you can see how the property parameters vary as different decisions are made during the process. Open Example Model > Model Simulation > BPSim Models > Car Repair Process.

The overall process is represented by this diagram:



The *Repair Car* Activity is a composite element that contains this sub process diagram:



Step	Action
1	In the Browser window, expand the 'BPSim' child Package underneath the 'Car Repair Process' Package, and double-click on the 'Scenario 1: Main Flow' Artifact. The Configure BPSim window displays. On the 'Scenario' tab look at the 'Duration' field; this has been set to 2 days and 12 hours (that is, 60 hours).
2	In the process hierarchy on the left of the window, expand the 'Start Event' category and click on 'Customer Arrives'. Select the 'Control' tab and look at the 'InterTriggerTimer' parameter, which has the value 24 minutes; that is, a customer arrives every 24 minutes (so over the 60 hour duration, 150 customers pass through the repair shop). Each customer enters the repair shop with one or more issues to be evaluated and repaired. The number of issues each customer presents can be randomly generated using one of the Distributions supported by BPSim. As the issue number is counted in discrete units (rather than measured on a continuous scale) we would use an 'integer' distribution. If you select the 'Properties' tab for the 'Customer Arrives' Start Event, you will see that the 'noOfIssues' property value is initialized from a Poisson distribution with a mean of 3.
3	Now expand the decision Gateway 'Accepted?' and its connectors, in the process hierarchy. 'Initial Estimate Accepted' has a Control parameter 'Probability', set to 0.67. The alternative connector, 'Initial Estimate Not Accepted' has a similar Control parameter 'Probability' set to 0.33. That is, we expect an average of one issue in three to be withdrawn - or not pursued - by the customer.
4	Further into the process, when an issue is being assessed on the vehicle, there is a possibility of another issue being discovered. In the list of Gateway elements, the last 'unnamed element' has two paths: 'New Issue Found' and 'No Added Issue Found'. Click on each of these and look at the 'Control' tab; the 'Probability' parameter for 'New Issue Found' is set to 0.25 and, for 'No Added Issue Found', to 0.75. So on average, for every four issues reported and assessed, one new one is discovered. The 'New Issue Found' path takes the process to the 'Handle New Found Issue' Activity, which adds 1 to the number of issues to be processed for the current customer. Expand the Activity group, and click on the 'Handle New Found Issue' element and on the 'Property' tab. You will see that the property 'noOfIssues' here has the Expression value {noOfIssues} + **1**.
5	When a problem with the vehicle is resolved, the 'Repair Issue' Activity deducts 1 from the number of issues to be repaired for the current customer. Click on the 'Repair Issue' element in the Activity group and on the 'Property' tab. You will see that the property 'noOfIssues' here has the Expression value {noOfIssues} **- 1**.
6	The value from the 'Repair Issue' Activity is tested at the 'Have Further Issues?' Gateway. Click on the 'More Issues to Repair' connector and on the 'Control' tab; the Condition parameter for following this path is set to the Expression value {noOfIssues} **> 0**; flow passes to the Gateway prior to the 'Inspect for Issue' Activity. Similarly, if you click on the 'No More Issues to Repair' connector and on the 'Control' tab, the Condition parameter for following that path is set to the Expression value {noOfIssues} **=< 0**, and flow passes to the

	'Repairs Completed' End Event. Now that you have examined the process flow and configuration settings, you can run a simulation and review the results.
7	On the Configure BPSim window, click on the Run button, and then on the 'BPSim Simulation Controller' dialog click again on the Run button, selecting the 'Standard' simulation (however, the type of simulation makes no difference for reviewing property parameters). On the 'BPSim Simulation Controller' dialog, you can review the Token status (and see that an additional customer manages to enter the shop at the very last minute) but it is difficult to see exactly how this summary data resulted. Click on the Attributes button to obtain the detailed property parameter values information on the 'BPSim PropertyParameter Values' dialog.
8	On the left hand side of the dialog is a summary of the minimum and maximum values for the property parameter (attribute) for each element in the process. For example, for the 'Customer Arrives' element the 'noOfIssues' parameter has a minimum of 0 and a maximum of 8, as generated by the Poisson (3) distribution. In the 'Token Number' field, type in a number (N) between 0 and 150 to select for the Nth customer who entered the repair shop. Click on the Query button to obtain the property parameter values used in the process for that customer. Review the results on each of the two tabs: • On the 'Group by Element' tab, see how the attribute's value changes in each element; for example, for customer 24 the 'noOfIssues' parameter is initialized with a value of 4 by the random distribution, and the 'Inspect for Issue' Activity is called six times with the parameter value being adjusted to 3 for three of those calls before cycling to 1, and the 'Handle New Found Issue' Activity is called twice with the parameter value at 3 both times • On the 'Group by Property' tab see how the parameter value changes as the process cycles through the Activities to completion, starting at 4, being adjusted between 3 and 4 a number of times and then decrementing to 0 at the end
9	Continue to explore the results as required, selecting different customers (Tokens). You can also return to the BPSim configuration and change the parameter initializations and add new ones, or change the decision points, to experiment with the process.

Responding to real numbers in the simulation of an integer-based process

In some cases, you might need to generate property parameter values using a distribution that returns 'real' numbers when the activities in the process operate with integers, or when you want to see what impact forcing integer values has on the process.

One mechanism to apply in such cases is to set conditions to avoid absolute numbers. So, for example, you might have a counter that decrements by 1, that is initialized to a 'real' number. If you set a condition to 'value==0' (equals 0) or 'value !=0' (does not equal 0), the two conditions might never be True or might always be True, respectively, causing an infinite loop. To avoid that, in the conditions you would use operators such as:

'value > 0'

'value < 0'

'value >= 0'

'value <= 0'

Another mechanism is to edit the code template used by the BPSim engine, to intercept and replace the real numbers provided to specific parameters with integers, as shown:

1. Select the 'Develop > Source Code > Options > Edit Code Templates' ribbon option.
2. In the Code Template Editor, in the 'Language' field, click on the drop-down arrow and select 'MDGBPSimExecutionEngineExtension'.

3. In the list of (Java) templates, click on 'MDGBPSimExecutionEngineExtension Compute Value'. The template contents display in the 'Template' panel.
4. Find this line:

   ```
   double %bpsimPropertyParameterName% = (double) distribution.next();
   ```

   Change it to:

   ```
   %if bpsimPropertyParameterName == "noOfIssues" or  bpsimPropertyParameterName == "noOfVisitors"%
   double %bpsimPropertyParameterName% = (int) distribution.next();
   //double %bpsimPropertyParameterName% = Math.ceil(distribution.next());
   //double %bpsimPropertyParameterName% = Math.floor(distribution.next());
   //double %bpsimPropertyParameterName% = Math.round(distribution.next());
   %else%
   double %bpsimPropertyParameterName% = (double) distribution.next();
   %endIf%
   ```
5. Replace the property parameter names with your own property parameters.
6. Click on the Save button, close the Code Template Editor and reload the project.

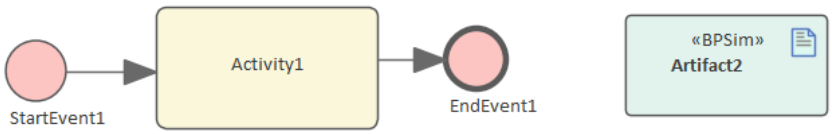
As presented, for each specified parameter the code template will simply replace any 'real' number initialized by the distribution with an integer. If you prefer, you can use one of the commented lines instead:

- Math.ceil() will take the 'real' number and convert it to the next highest integer
- Math.floor() will take the 'real' number and convert it to the next lowest integer
- Math.round() will take the 'real' number and round it up or down depending on whether it is greater than or less than n.5

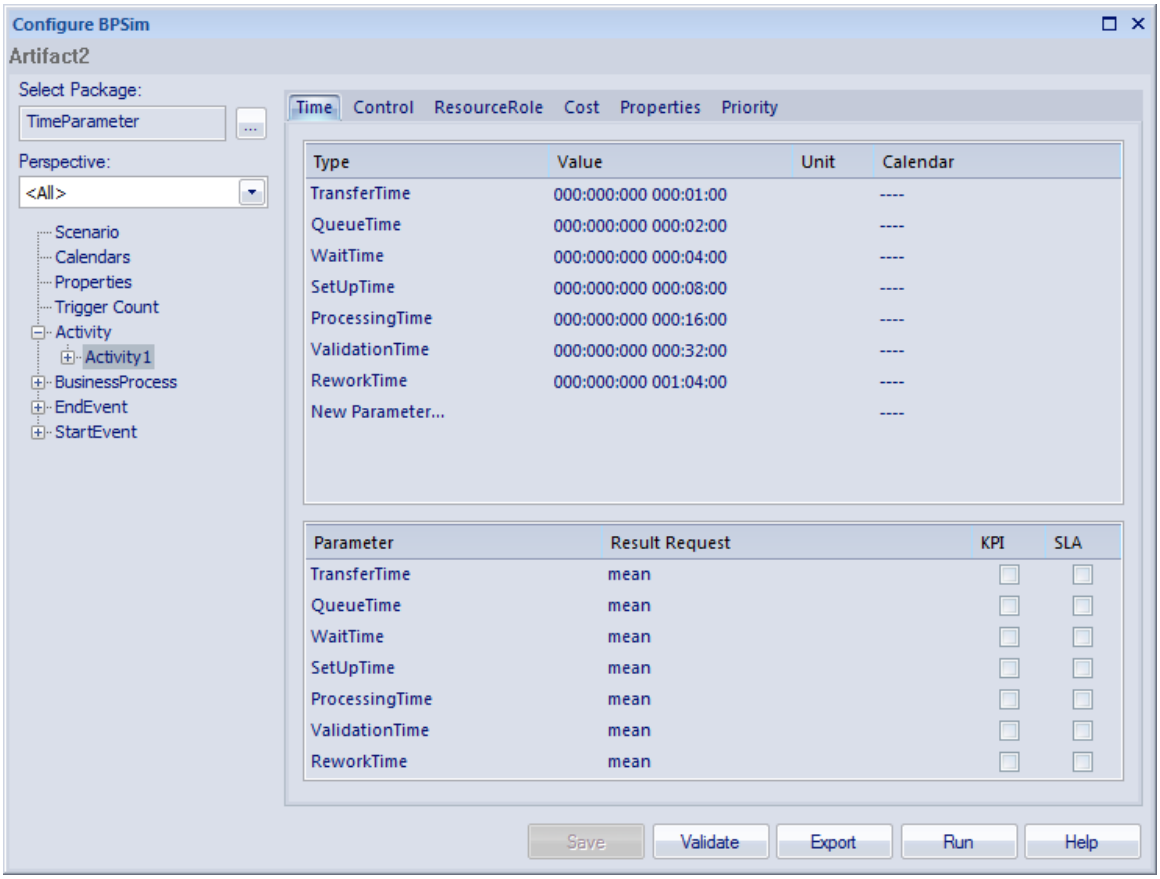
Time Parameter Behavior

In the BPSim configuration, you can set a number of Time parameters for an Activity, such as Queue Time and Wait Time. You can also set a Result Request on each of these, for a Custom simulation. However, the BPSim simulation engine combines these parameters into a single 'Processing Time' quantity.

Consider the simple model TimeParameter in the Example Model (Example Model > Model Simulation > BPSim Models > Time Parameter), represented by this diagram:

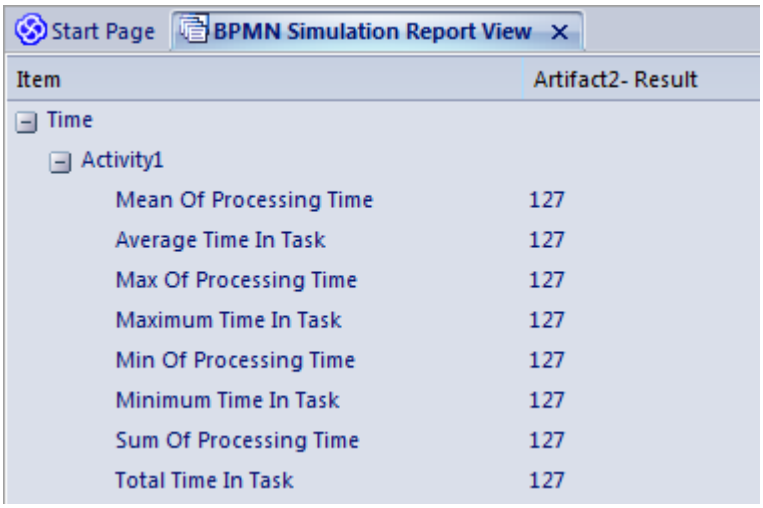


If you double-click on the Artifact2 element, the Configure BPSim window displays. Click on the Activity1 element in the diagram to expand the Activity group, to select Activity1 in the hierarchy at the left of the dialog, and to display the first tab, 'Time', for the element in the configuration, as shown.



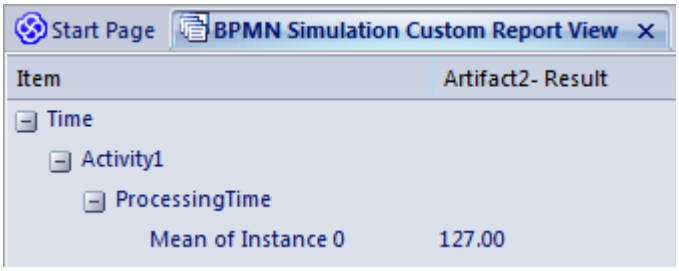
Note that in the upper panel there are seven system-supplied 'Time' parameters, which have been given initial values of - in order - 1, 2, 4, 8, 16, 32 and 64 minutes (the 64 minutes is 1 hour and 4 minutes). Note also that in the lower panel, each of these has a Result Request for the mean runtime value of the parameter.

Click on the Run button, and on the 'BPSim Simulation Controller' dialog click on the Run button and select 'Standard Simulation'. The simulation is configured to cycle through the process once. When the simulation is complete, click on the Open Result button, and on the 'BPMN Simulation Report View' right-click and select the 'Show Only Non-Empty Items' option. This gives you, for the Activity1 element on which the parameters were set, these results:



All of these derived results are 127 minutes, the sum of the initial values of the original seven 'Time' parameters. The individual parameters are not processed separately.

If you return to the 'BPSim Simulation Controller' dialog and click on the Run button, selecting 'Custom Simulation' this time, the Open Result button displays the 'BPMN Simulation Custom Report View'. In the configuration, the Result Requests were for the mean values of the seven parameters. In the Report View for the simulation, you only see the mean of the single aggregated parameter, ProcessingTime, as 127 minutes.


Start Page
BPMN Simulation Custom Report View
Item
Artifact2- Result
Time
Activity1
ProcessingTime
Mean of Instance 0
127.00

Compare BPSim Configurations

When you develop a BPSim configuration, you can define a wide range of parameters to set prior to running simulations and observing the effects both of those settings and of changes in selected settings. To facilitate using and managing multiple 'what-if' scenarios, it is recommended that you create copies of the original configuration (as Artifact elements) and make the setting changes in the copies.

A useful facility in creating variations of a configuration is to apply inheritance, whereby the data and parameters you do NOT intend to vary are held in one configuration, and only those parameters that you change are held in another. The 'variable' configuration uses (inherits) the common data held in the base configuration, so you do not have to re-create that common data in the 'variable' configuration.

You can then run simulations on the changed configurations and on the original 'baseline' and compare the simulation reports to see what differences occurred in the run-time variables, and then run and display the comparisons of the configurations to see what changes in parameter settings gave rise to those run-time differences.

By running simulations under the original and copy configurations, comparing the results and the changes that caused the results, and modifying the model accordingly, you can achieve a very high degree of control in streamlining the business process you are developing.

Access

Context Menu	In a diagram or the Browser window | Right-click Business Process Simulation Artifact | Show BPSim Configuration > Review tab > Configuration Summary tab

Configuration Summary Tab

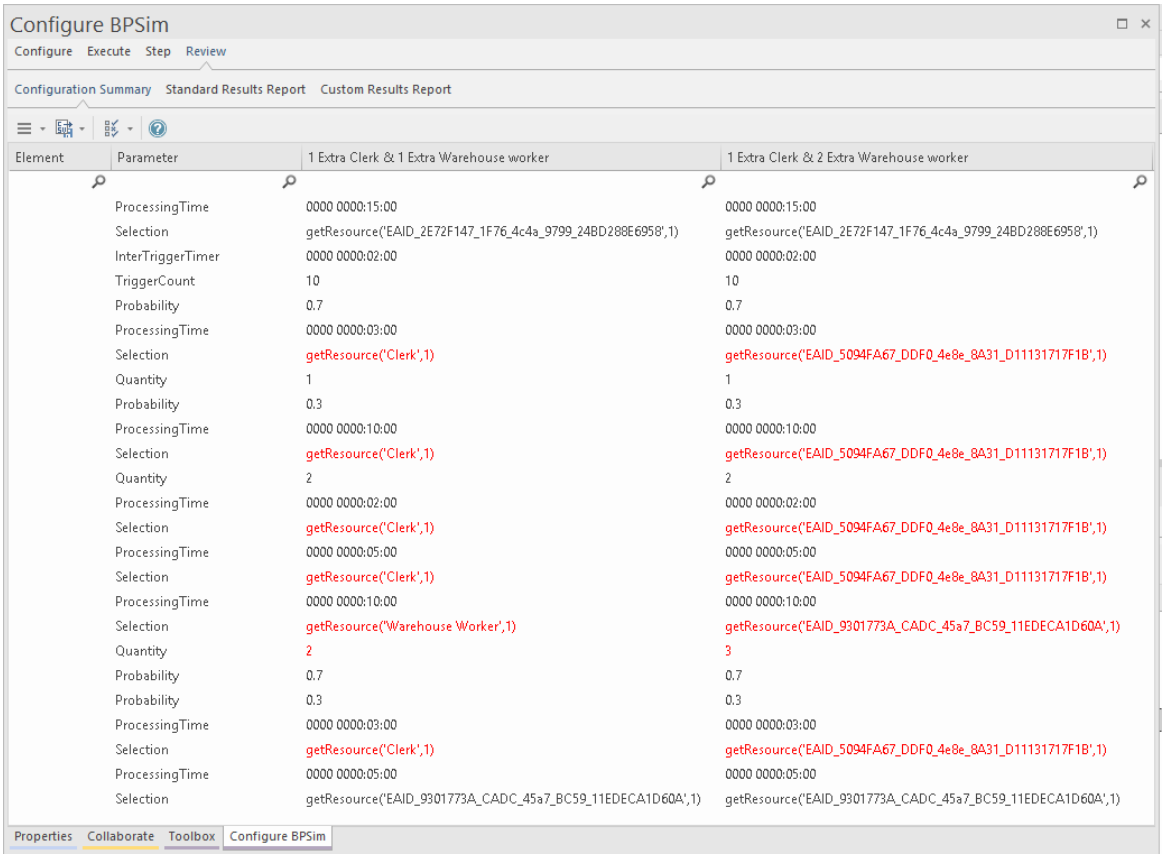
This view initially displays the parameters in the selected configuration that have values and, in the column under the configuration name, the values set for those parameters. If the window is docked you can review the results more easily by dragging the tab off the workspace to become a floating display, and expand the display to full screen size.

To compare two (or more) configurations, click on another BPSim Configuration Artifact element in the Browser window or diagram and either:

- Drag it onto the report view or
- Select the 'Show BPSim Configuration' option

You can also click on the icon on the 'Configuration Summary' tab, click on the 'Add BPSim Configuration' option and browse for and select the Artifact element on the 'Select Element' dialog.

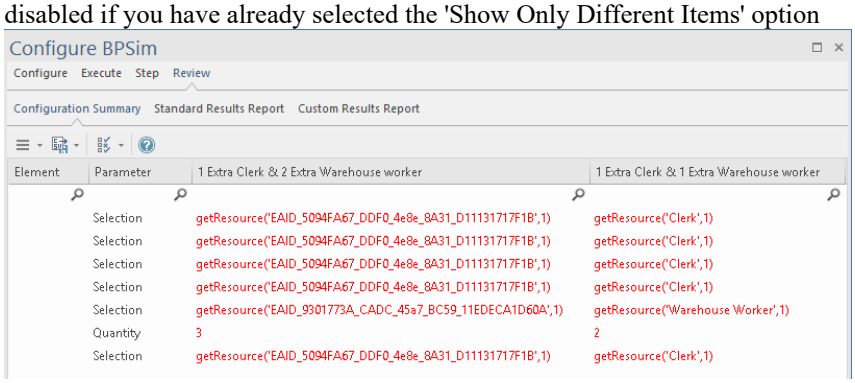
The parameter hierarchy now contains any additional parameters in that configuration, and its parameter values display in a column to the left of the original configuration values, with the name of the Artifact in the column header.



You can review and manipulate the information on the report using options available from the toolbar and the right-click context menu.

BPMN Simulation Configuration Summary Options

Option	Description
	Select this toolbar option to display a short menu of options: • Add BPSim Configuration - displays the 'Select Element' dialog, through which you can browse for and select another BPSim Artifact and display its parameters next to those already on the page • Reload - refresh the display and return column changes to the original widths and positions • Clear List - clear the selected Artifacts and their parameters from the display • Remove Report <name> - clears the named Artifact and its parameters from the page; one of these options is provided for each Artifact currently shown on the page
	Click on this toolbar icon to export the records on this page to a CSV file or XML file. The 'Save As' browser displays, from which you select the location into which to save the file.
	Displays two options for filtering the information displayed on the page; the options are enabled when you have two or more configurations shown: • Show Only Different Items - select this option to restrict the display to only those parameters where the values differ between the configurations; click on

	the option again to deselect it • Highlight Different Items - (when two or more configurations have different parameter values) displays the differing parameter values in red; this option is disabled if you have already selected the 'Show Only Different Items' option 

BPSim Charts

The 'Charts' page of the Diagram Toolbox provides two icons specifically to generate Charts that reflect selected results from BPSim simulations. These are:

- BPSim Result Chart - to generate a Chart that reflects selected results from a series of standard BPSim simulations
- BPSim Custom Result Chart - to generate a Chart that reflects results from a series of customized BPSim simulations

As for other Chart Artifacts, both BPSim Chart types can be quickly configured to display the simulation results in variations of a Time Line chart, 2-dimensional Bar Chart or 3-dimensional Bar Chart.

Prerequisites

To populate the Charts created from the Business Process Simulation Artifacts, you select the Result Artifacts created during the simulation of each configuration that you want to show. Therefore, the initial simulations must be performed first, and the Report Artifacts generated.

Access

Display the 'Charts' page of the Diagram Toolbox using any of the methods outlined in this table.

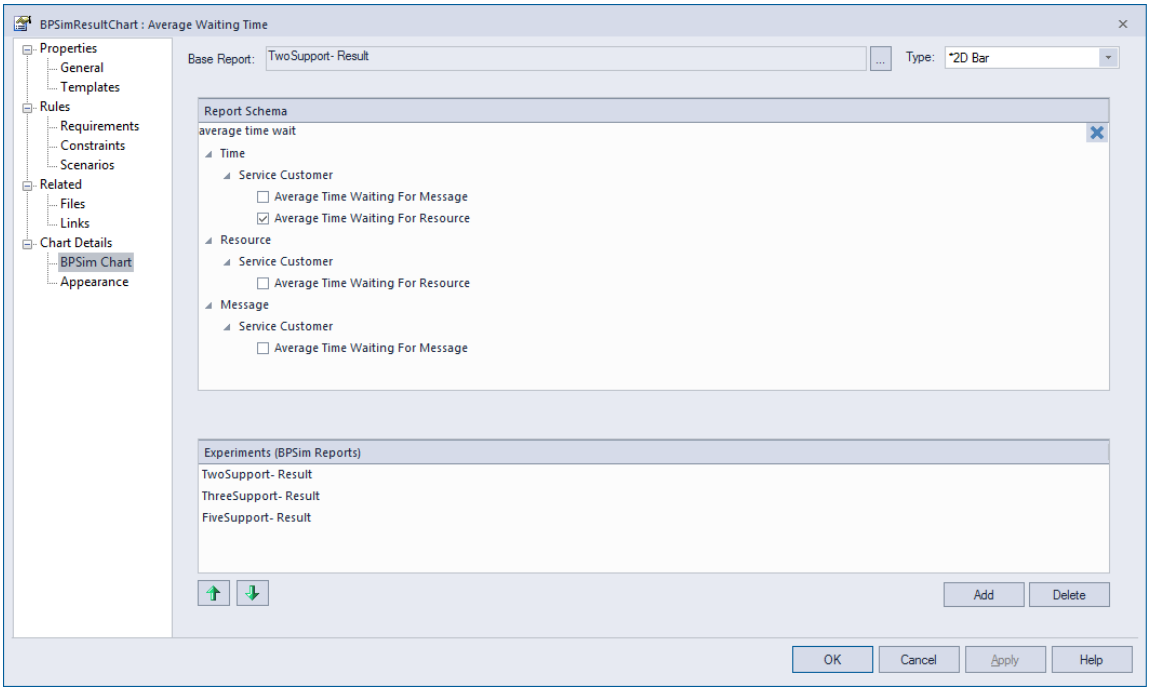
Then, drag the BPSim <type> Chart Artifact icon onto the diagram - a new Chart element is created.

Double-click on the new Chart element to open the 'Properties' dialog, showing the 'BPSim Chart' page.

Ribbon	Design > Diagram > Toolbox > Charts
Keyboard Shortcuts	Ctrl+Shift+3 |Charts
Other	You can display or hide the Diagram Toolbox by clicking on the » or « icons at the left-hand end of the Caption Bar at the top of the Diagram View.

Select Results to Display In Chart

Complete the fields on the 'BPSim Chart' page of the Chart 'Properties' dialog.



Option	Action
Base Report	Click on the ... button to display the 'Select <<BPSimReport>>Artifact' dialog, and select the report Artifact for the simulation results against which other simulation results will be compared. This Result Artifact is also added to the 'Experiments' panel as the first in the list of report results to be compared.
Type	Click on the drop-down arrow and select the format type of the Chart in which to display the results - 2D Bar, 3D Bar or Line. After you have specified the report parameters to compare, you can select the 'Appearances' page of the dialog and define the appearance of the Bar Chart or Time Line graph.
Report Schema	Expand the hierarchy as necessary and select the checkbox against each property to display on the Chart. Each property will be represented by a separate line or group of bars on the Chart. Usually you would select similar objects (such as different resources) and the same single property for each object (such as the degree of utilization of the resource). You have a wealth of properties to examine and compare, but any more than a couple on the same Chart makes the Chart hard to read.
Experiments (BPSim Reports)	This panel lists the simulation report results (as BPSim Result Artifacts) that you have selected to compare using the Chart, in the sequence in which their selected parameter will be shown on the Chart. Usually, the Base Report remains as first in the list and the result for its parameter is shown at the left of the Chart. If you want to change the sequence, click on the Result Artifact name and click on the appropriate Up/Down green arrow button. To add further Result Artifact names to the list, click on the Add button and browse for and select the Artifact from the 'Select <<BPSimReport>> Artifact' dialog.

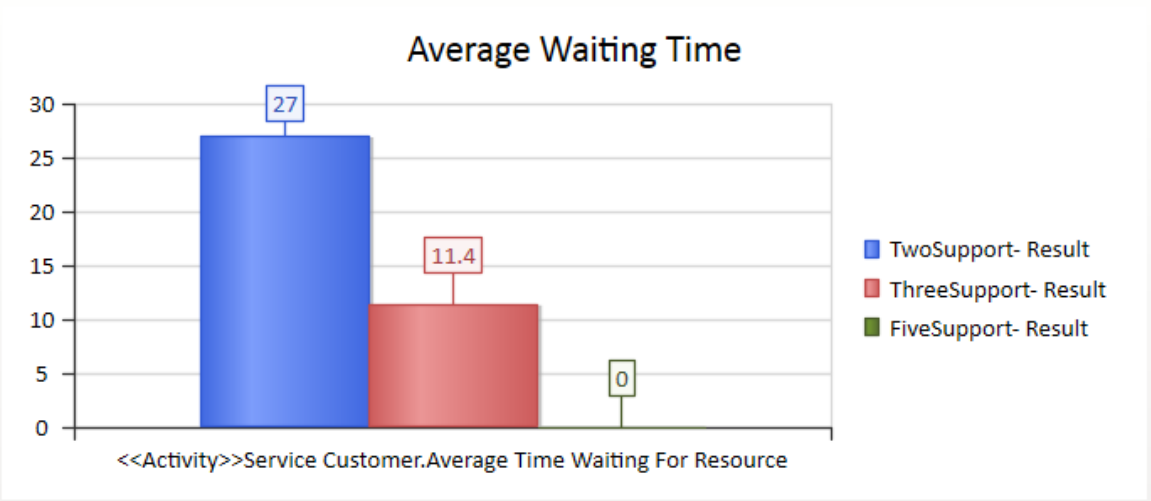
A BPSim Chart Example

In the Help Desk Phone Support example, we created three BPSim Artifacts to do a what-if analysis of 'How many support resources do we need to answer the customer's questions over the phones in an economical way?'

We started with two support resources, then tried three and five resources. After simulation, we have a BPSim Report based on different configurations: TwoSupport-Result, ThreeSupport-Result and FiveSupport-Result.

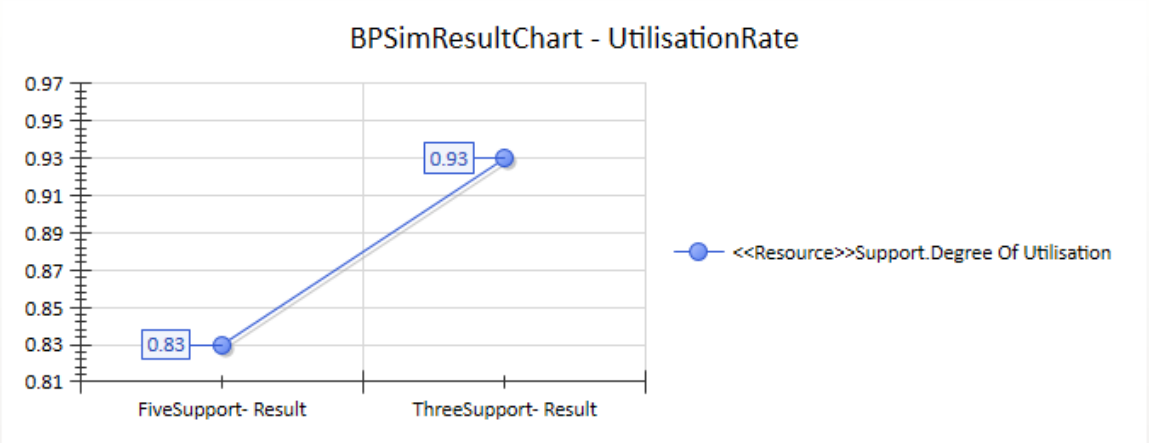
These are the steps to create a Chart to compare the customer's average time waiting for support:

1. Create a BPSim Result Chart on the diagram, and name it *Average Waiting Time.*
2. Double-click on the Chart to open the Properties window, then open the 'BPSim Chart' tab.
3. Click on the ... button to select a Base Report, from which we define the schema (legends) to use in the Chart. Select 'TwoSupport-Result'.
4. Choose this schema:
   - Time | Service Customer | Average Time Waiting For Resource
5. Click on the Add button to add another two BPSim Reports: 'ThreeSupport-Result' and 'FiveSupport-Result'.
6. Click on the OK button, and adjust the size of the Chart element. This Chart gives us direct information.



These are the steps to create a Chart to compare the degree of use of Support:

1. Create a BPSim Result Chart on the diagram, and name it *Utilisation Rate.*
2. Double-click on the Chart to open the Properties window, then open the 'BPSim Chart' tab.
3. Click on the ... button and select a Base Report, from which to define the schema (legends) to use in the Chart. Select 'TwoSupport-Result'.
4. Choose this schema:
   - Resource | Support | Degree Of Utilisation
5. Click on the Add button to add another two BPSim Reports: 'ThreeSupport-Result' and 'FiveSupport-Result'.
6. Click on the OK button and adjust the size of the Chart element. This Chart provides specific information.


BPSimResultChart - UtilisationRate
0.97
0.95
0.93
0.91
0.89
0.87
0.85
0.83
0.81
0.93
0.83
FiveSupport- Result
ThreeSupport- Result
<<Resource>>Support.Degree Of Utilisation

BPSim Examples

This section provides several examples of BPMN modeling, BPSim Configuration and analysis of simulation results.

These examples can all be accessed from the EAExample model.

To run the example simulations you must have the BPSim Execution Engine and Java Runtime Environment (JRE) 1.7 or higher installed. To work with Property Parameters you must also have Java Development Kit (JDK) version 1.7 or higher installed.

Meal Order Collaboration Version 1

In this example, we create a very simple model to simulate the communication between a customer and a restaurant for a meal order.

For the customer's process:

1. A customer sends a message to the restaurant to order a meal.
2. The customer will wait for the delivery.
   If the delivery is not fulfilled within 60 minutes, they will call the restaurant, and then continue to wait.
3. Upon the delivery, the customer will have dinner.

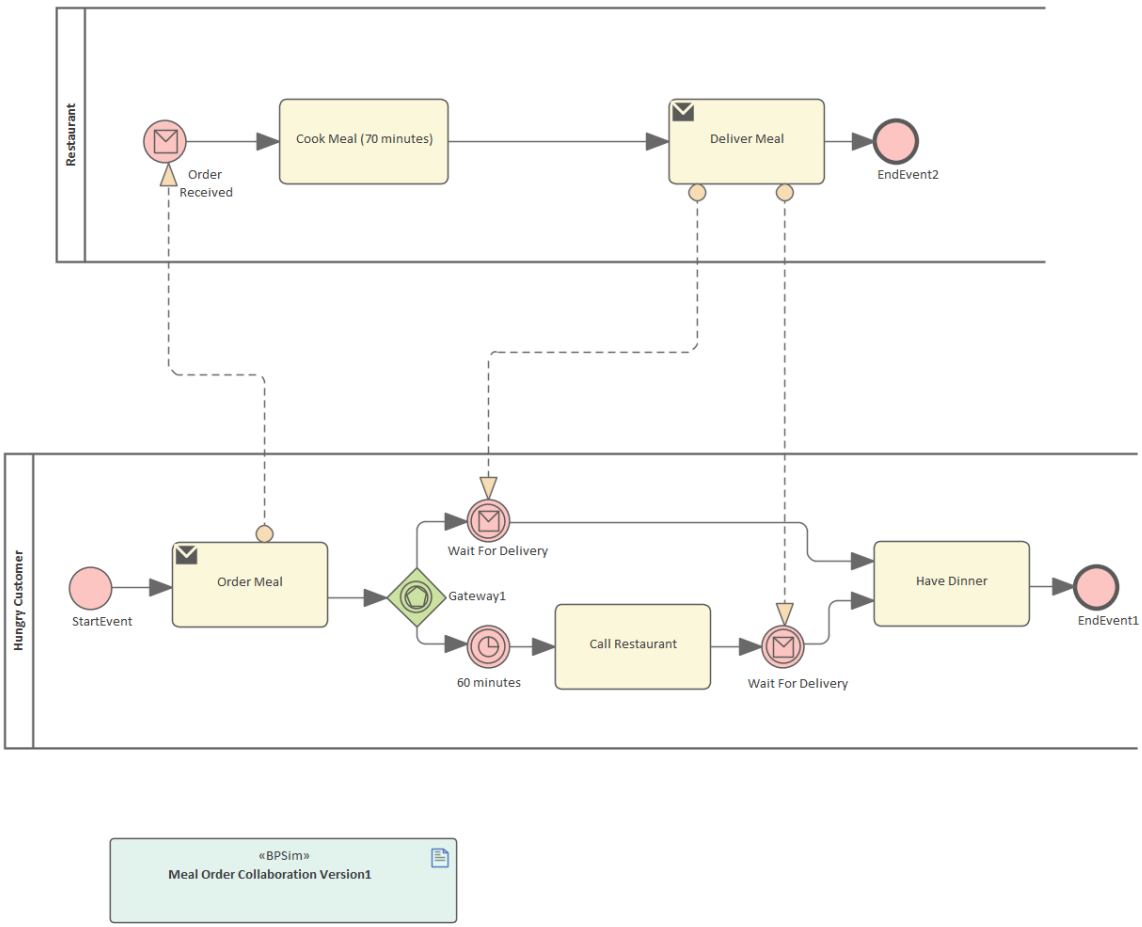
For the restaurant's process:

1. The process is started on receiving a meal order from the customer.
2. The cooking time can be set by the user. This allows experimentation with different events times, e.g. 30 minutes, 70 minutes
3. The restaurant delivers the meal and finishes the process.

Create BPMN Model

To set up a BPMN model that can be used for this BPSim simulation you:

- Create a Collaboration Model with 2 Pools
- Within each Pool create an element for each process
- Connect the elements with Message Flows for the process communication
- Include a BPSim Artifact for setting the simulation details.

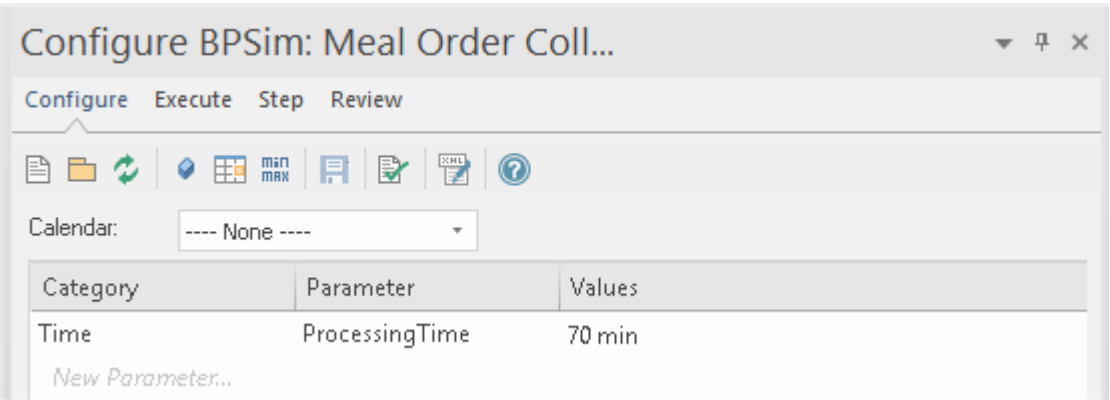


Configure BPSim

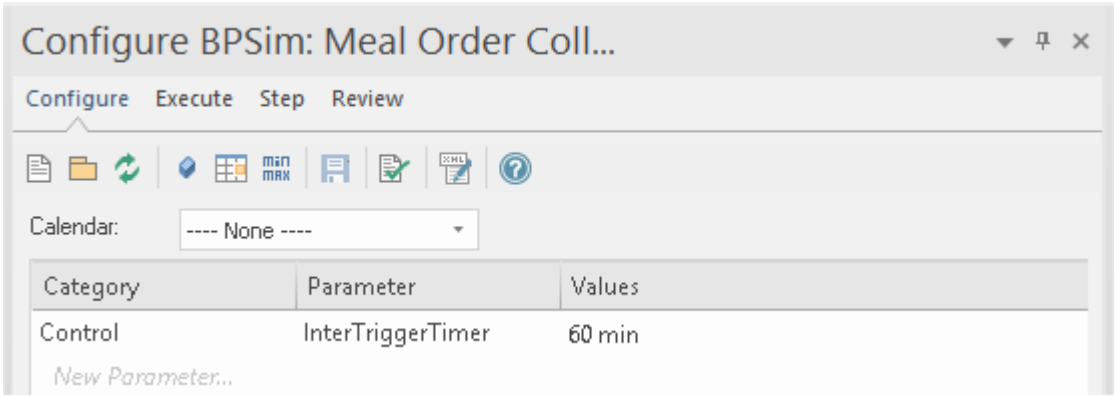
In this example we configure the following BPSim parameters:

- Set the TriggerCount of StartEvent in Hungry Customer to 1

- Set the ProcessingTime of Cook Meal to 70 minutes



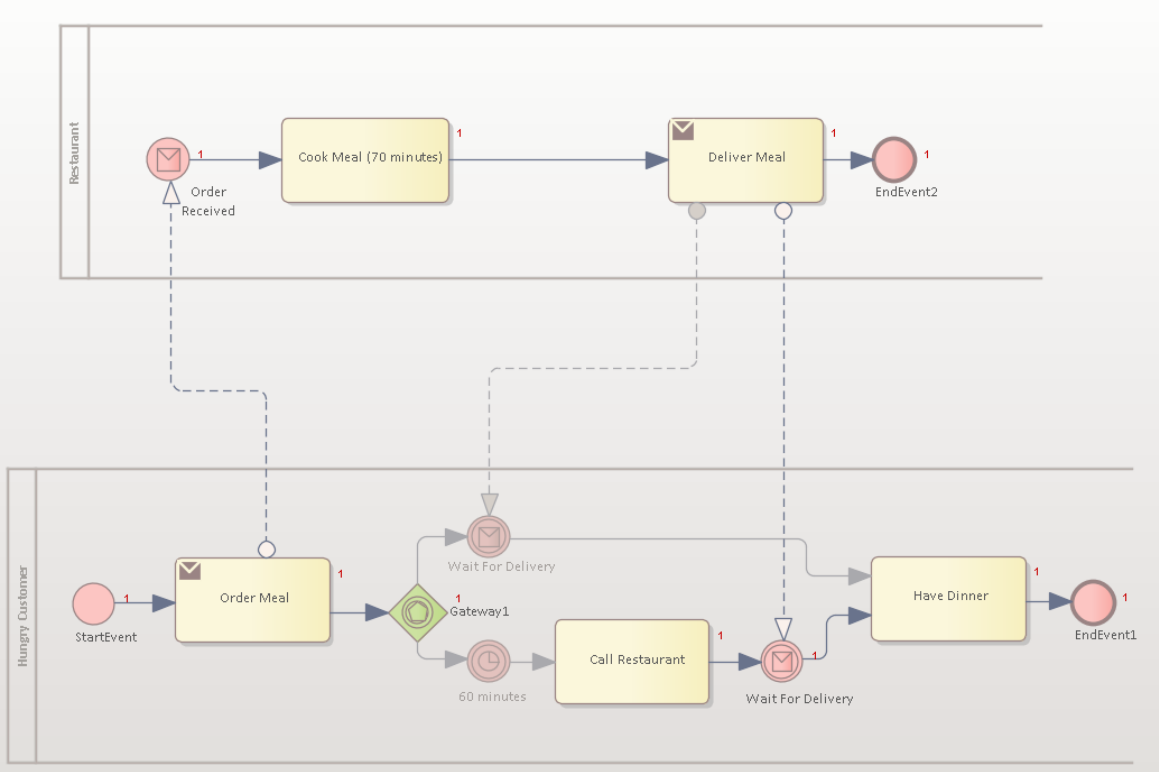
- Set the InterTriggerTimer of Intermediate Event to 60 minutes



- Default settings on other BPSim parameters, here is a list of configurations, you can view thourgh Review > Configuration Summary

Simulation

Ensure the Config BPSim window is open (Simulate > Process Analysis > BPSIM > Open BPSIM Manager).
Navigate to Execute tab and run the Standard Simulation:



The Exclusive Event Gateway was triggered by the 60 minutes Timer event when the Cooking Meal task took 70 minutes.

If we change the BPSim setting for Task: *Cook Meal*: ProcessingTime from 70 minutes to 30 minutes, the Exclusive Event Gateway will be triggered by the message event *Wait For Delivery* and *Call Restaurant* task will not be activated at all.

Meal Order Collaboration Version 2

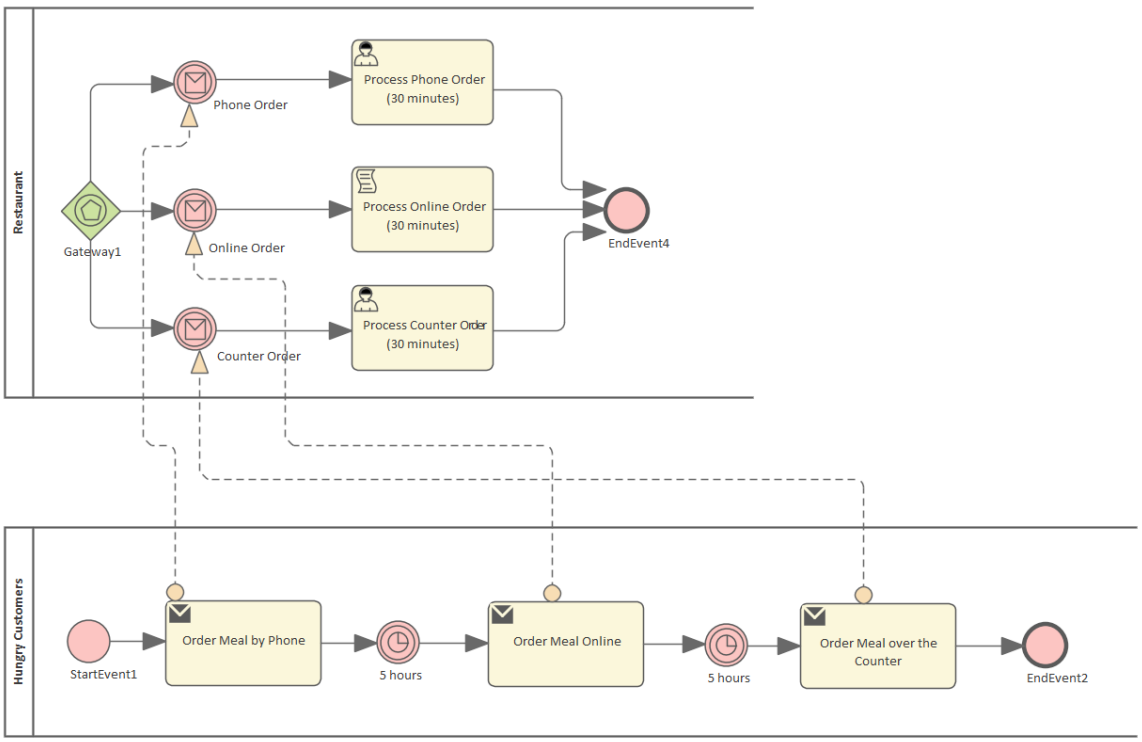
In this example, we create a model to simulate the communication between a customer and a restaurant for a meal order. The customer places three different orders by: phone, online and over the counter.

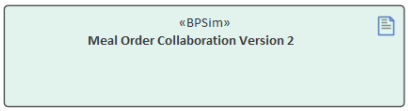
Create BPMN Model

To set up a BPMN model that can be used for this BPSim simulation you:

- Create a Collaboration Model with 2 Pools
- Within each pool create an element for each process
- Create Message Flows depicting the Process communication
- Include a BPSim Artifact for setting the simulation details.

Please note: the Event-Exclusive gateway's *instantiate* attribute was set to **true**, which means the process was started by this *Start Event Gateway*.

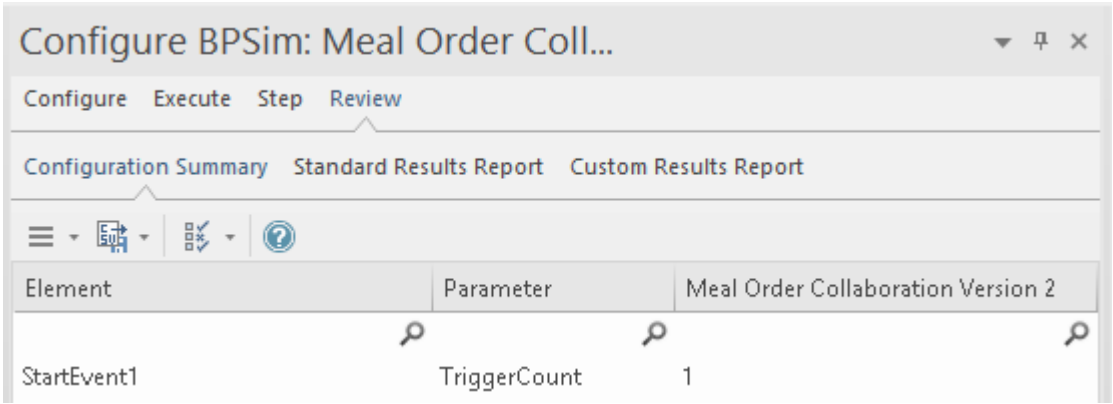




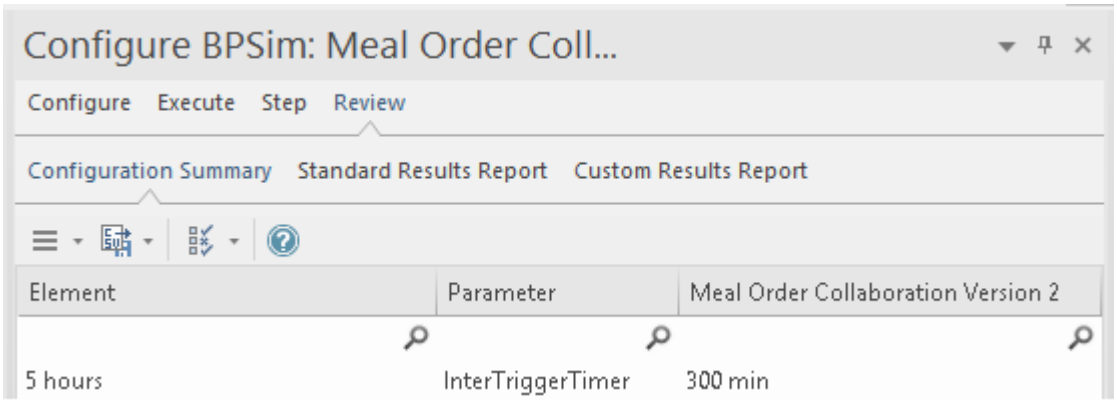
Configure BPSim

In this example we configure the following BPSim parameters:

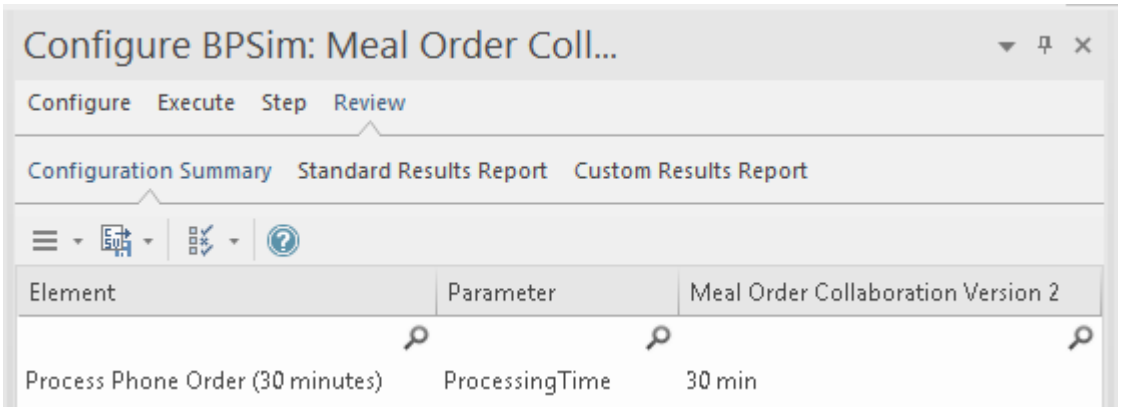
- For the Start Event, set the parameter TriggerCount to value 1



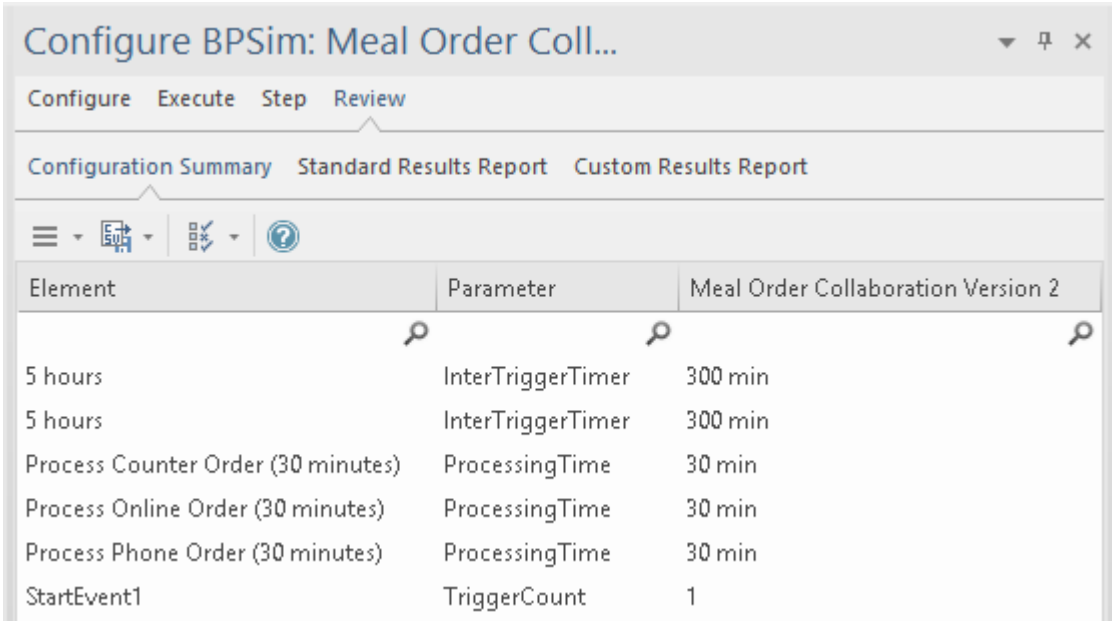
- For two of the Intermediate Event 5 hours, set the parameter InterTriggerTimer to value 300 min



- For three of the tasks in Restaurant, set parameter ProcessingTime to value 30 min

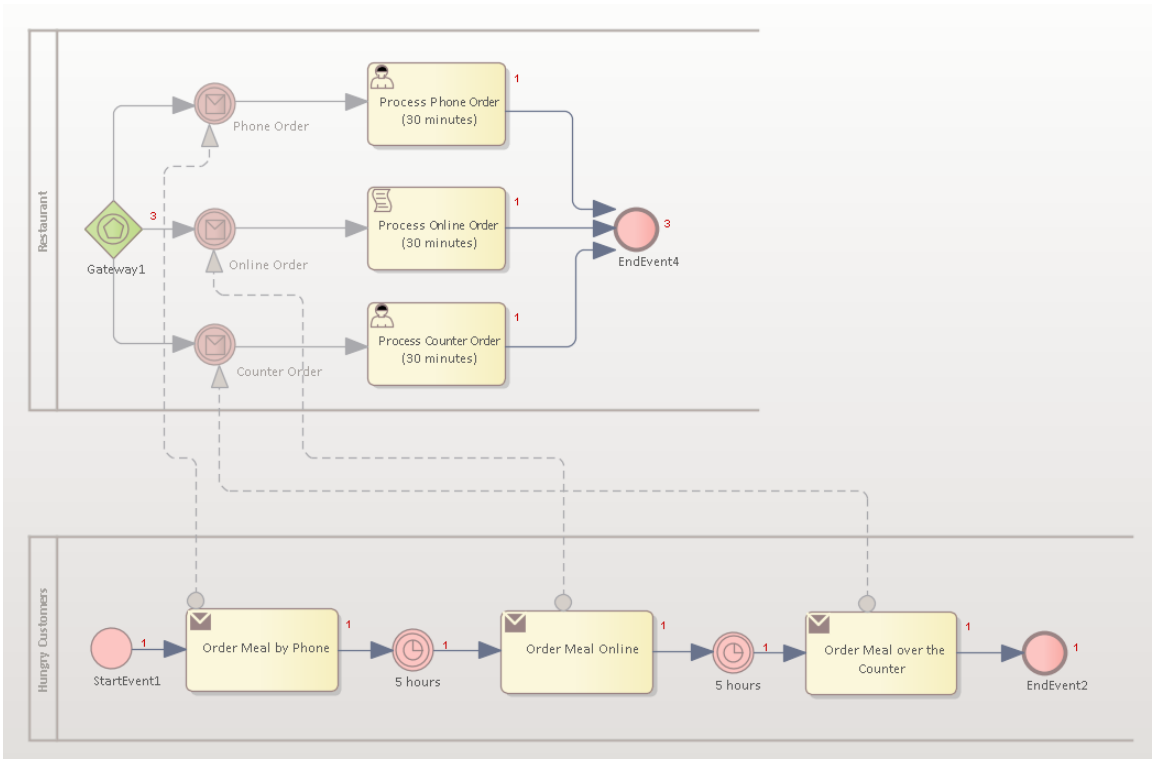


- Default settings on other BPSim parameters, here is a list of configurations, you can view thourgh Review > Configuration Summary



Simulation

Navigate to Execute tab and run the Standard Simulation:



The *Restaurant* process was instantiated 3 times by the Event-Ecxlusive gateway through 3 different messages.

Help Desk Phone Support Simulation

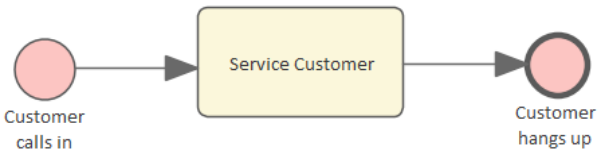
In this example, we create a very simple model to simulate a Help Desk phone support process.

We set up a scenario in which resources are limited and the requests have to be put in a waiting queue for a resource. Then we try to seek a balance point between a customer's waiting time and the number of resources, using a what-if analysis.

Firstly, we model this process step-by-step, starting with a simple parameter setting than can be calculated with pen and paper, then verifying it with BPSim. After that, we perform a what-if analysis that might help the manager to make a decision.

Create BPMN Model

The model itself is very simple, consisting of a Start Event, a Task and an End Event.



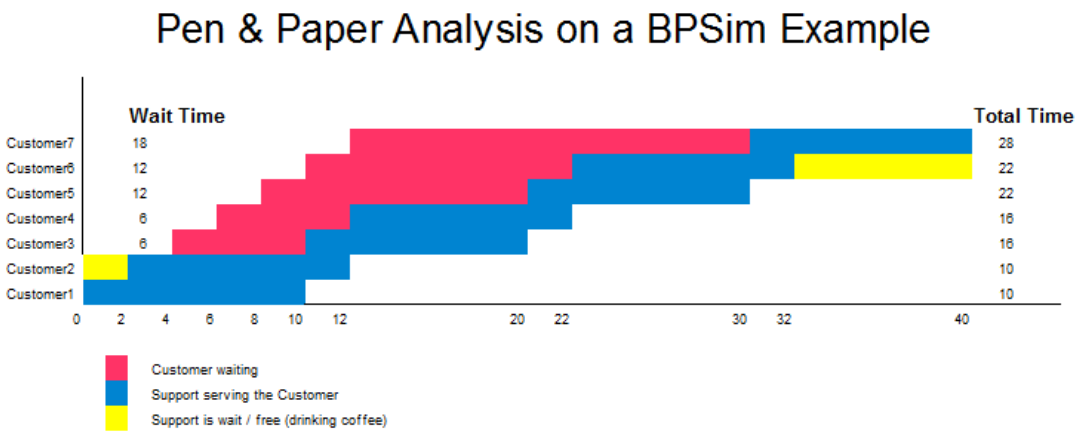
- Create a Start Event called *Customer calls in*
- Add a Sequence Flow to a target Abstract Task called *Service Customer*
- Add a Sequence Flow to a target End Event called *Customer hangs up*

Create a BPMN2.0 Resource named *Support;* this element will be used in the BPSim Configuration.

Pen and Paper Analysis

We will use pen and paper to analyze this case:

- 7 customers call in at 2 minute intervals
- 2 support resources are available
- Each service will take 10 minutes



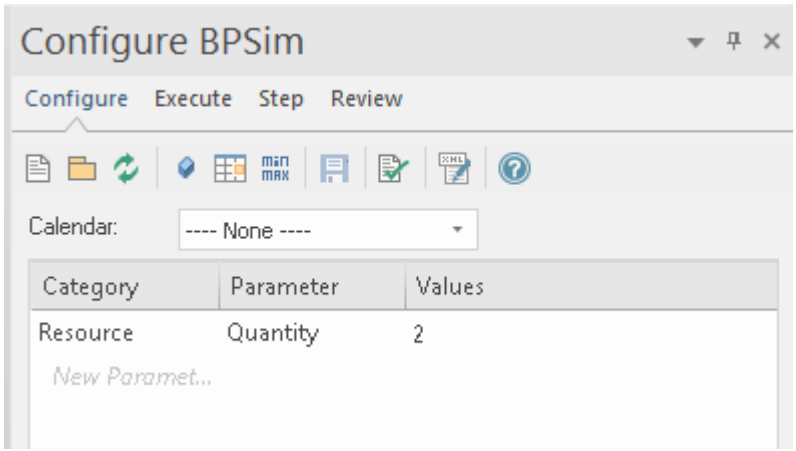
Now we can perform some calculations using a pen and paper based approach:

Total Time Waiting For Resource:	(18+12+12+6+6+0+0) = 54	number of
Total Time In Task:	(28+22+22+16+16+10+10) = 124	
Average Time in Task:	124/7 = 17.71	
Average Time Waiting For Resource:	54/7 = 7.71	
Maximum Time In Task:	28	
Maximum Time Waiting For Resource:	18	
Sum of Support's wait time:	2+8 = 10	number of
Sum of Support's time for Task:	7*10 = 70	number of
Degree of Utilisation:	70/(10+70) * 100% = 87.5%	

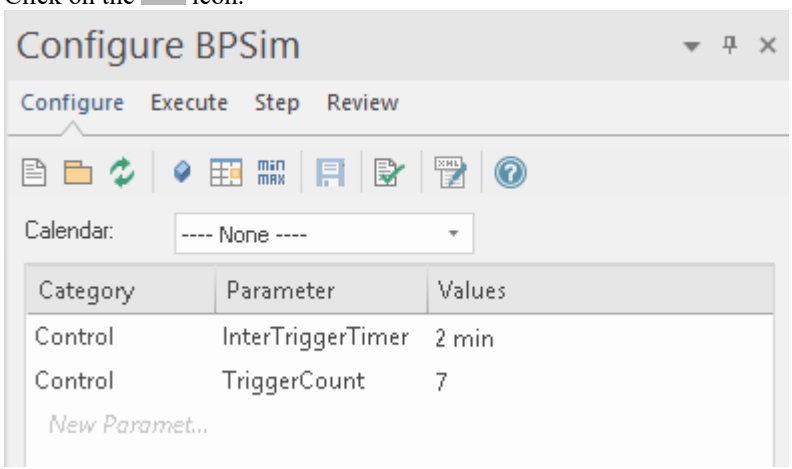
Seen from this result, it is already a very complicated computation for such a simple model when resource constraints are applied. When the process expands and more constraints are applied, analyzing the process with pen and paper will quickly become impossible. We will demonstrate how BPSim can help.

BPSim Configuration

1. Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager').
2. Click on the icon and the Add New button, and create a Business Process Simulation Artifact named *Pen & Paper Analysis 7 Customers.*
3. Click on the icon and browse for and select the Package containing the corresponding BPMN 2.0 model.
4. Open the model diagram and click on the Resource element called 'Support'.
5. In the 'Category' column on the window, click on the 'New Parameter' drop-down arrow and select 'Resource', then click on the 'Parameter' drop-down arrow and select 'Quantity', and in the 'Values' field type '2'.
6. Click on the icon.

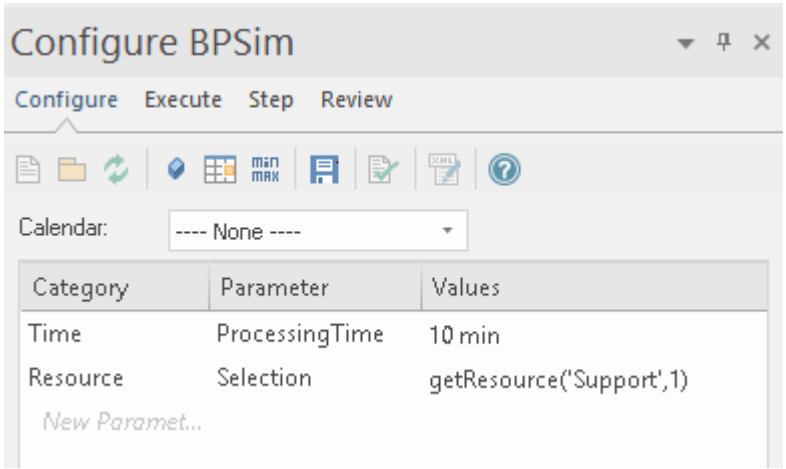


1. In the diagram, click on the 'Customer calls in' Start Event element.
2. In the 'Category' column in the window click on the 'New Parameter' drop-down arrow and select 'Control'.
3. Click on the 'Parameter' drop-down arrow and select 'InterTriggerTimer'.
4. In the 'Value' field, click on the ... button, select the 'Constant' tab and 'Numeric', type '2' in the 'Constant Numeric' field and select 'minutes' in the 'TimeUnit' field, then click on the OK button.
5. Repeat steps 2 and 3, selecting 'TriggerCount' in the 'Parameter' field, and in the 'Value' field type '7'.
6. Click on the icon.



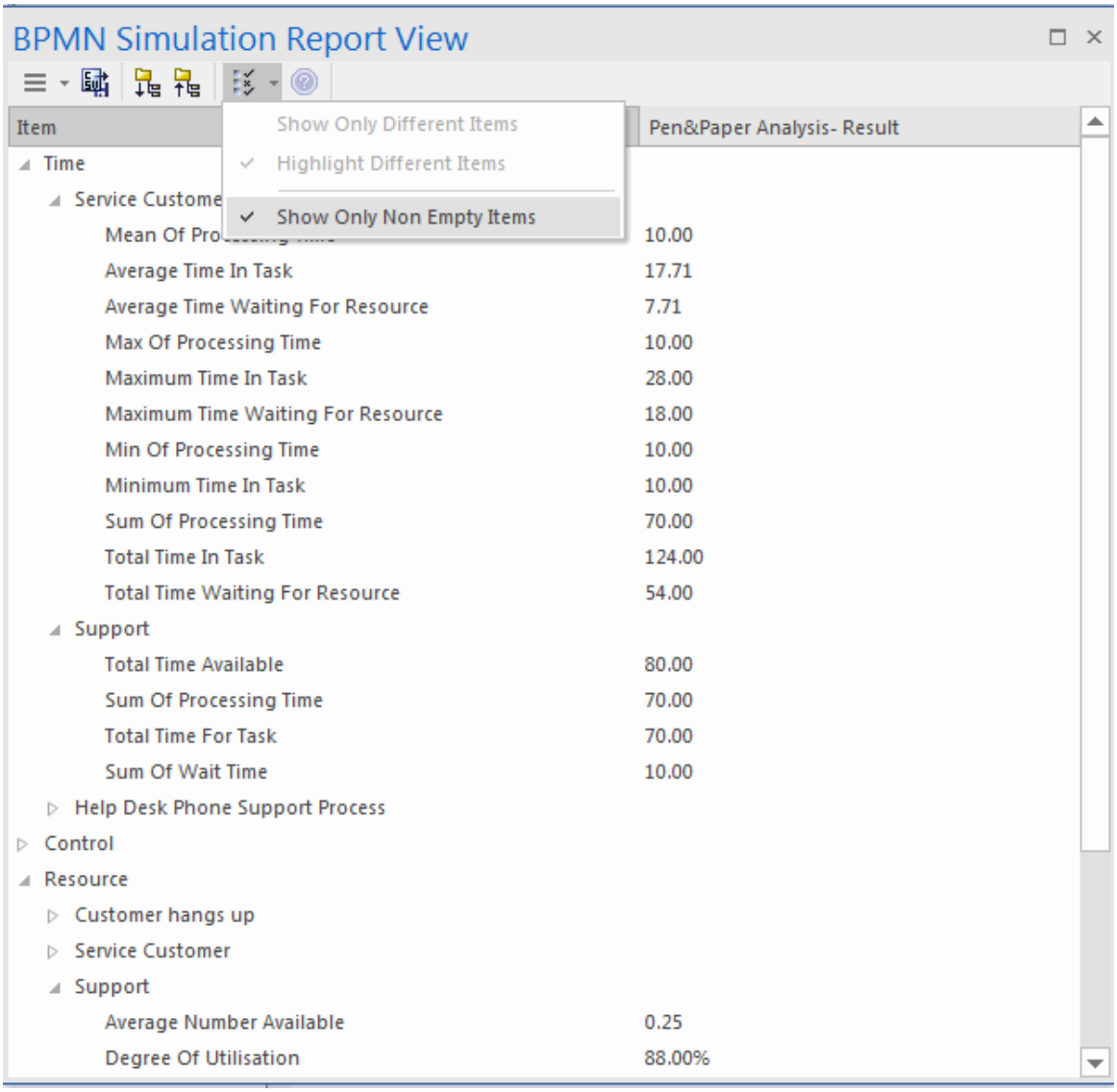
1. In the diagram, click on the 'Service Customer' Activity element.
2. In the 'Category' column in the window click on the 'New Parameter' drop-down arrow and select 'Time'.
3. Click on the 'Parameter' drop-down arrow and select 'ProcessingTime'.
4. In the 'Value' field, click on the ... button, select the 'Constant' tab and 'Numeric', type '10' in the 'Constant Numeric' field and select 'minutes' in the 'TimeUnit' field, then click on the OK button.
5. In the 'Category' column in the window click on the 'New Parameter drop-down arrow and select 'Resource'.
6. Click on the 'Parameter' drop-down arrow and select 'Selection'.
7. In the 'Value' field, click on the ... button, select 'Support' (the name of the Resource element you created in the model) and click on the Add Selection by Resource(s) button to move 'Support' into the 'Resource or Role' column.
8. In the 'Quantity Required' column type 1.
9. Click on the OK button. In the 'Values' field the automatically-generated expression *bpsim:getResource('Support',1)* displays.

10. Click on the icon.



Run the Simulation

1. On the Configure BPSim window click on the 'Execute' tab and on the icon in the toolbar.
2. When the simulation is complete, click on the icon to open the 'Configuration Summary' tab of the 'Results' tab'



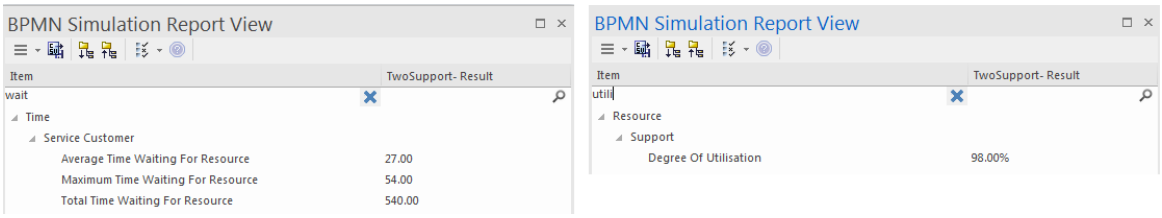
The results match the pen and paper analysis.

Simulation - 2 Support Resources for 20 customers

You can create a new Business Process Simulation Artifact by copying an existing BPSim Configuration. Copy the *Pen & Paper Analysis 7 customers* element and press Ctrl+Shift+V to paste, giving the new element the name *TwoSupport*.

1. Double-click on *TwoSupport* to open the Configure BPSim window; you can see that all the configurations are kept from the copied source
2. In the diagram, click on the 'Customer calls in' Start Event element.
3. In the 'Category' column in the window click on the 'Value' field for 'Control' - 'Trigger Count' and change the value to '20'.
4. Click on the icon.

Run simulation and analyze results



From the report, you can see that:

- The 'Average Time Waiting For Resource' is 27 minutes and the 'Maximum Time Waiting For Resource' is 54 minutes
- The two Support resources - are they busy? If they were not, we might have to change the process to use all of their time and reduce the customer's waiting time; however, the 'Degree Of Utilisation' is at 98%, which indicates that the resources had almost no idle time

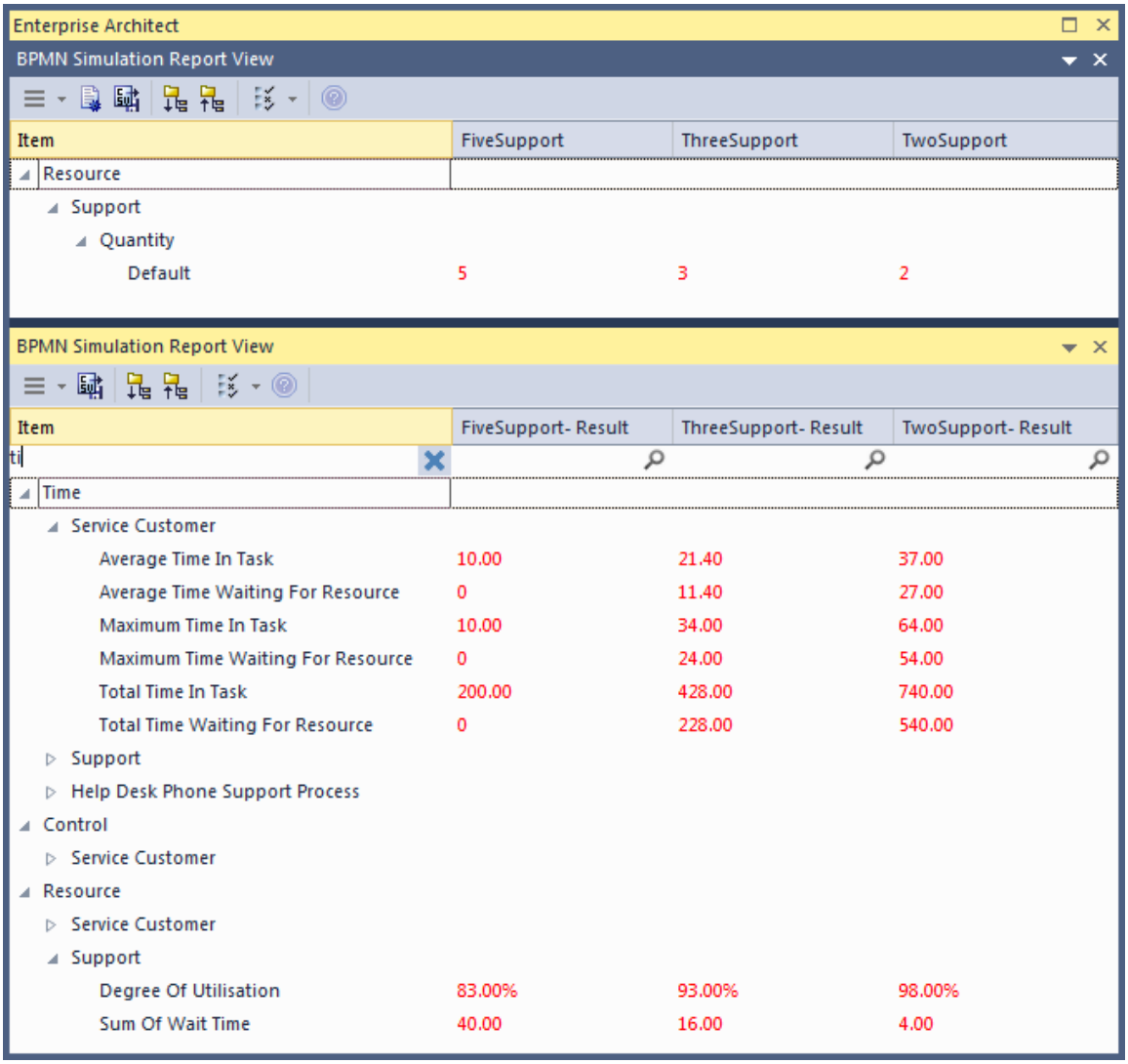
'What-If' I have more staff? Compare 2 Support Resources with 3 and 5 Support Resources

1. Copy *TwoSupport* and press Ctrl+Shift+V to paste, giving the new element the name *ThreeSupport.*
2. Double-click on *ThreeSupport* to open the 'Configure BPSim' dialog.
3. In the diagram, click on the 'Support' Resource element.
4. In the Configure BPSim window, in the 'Values' field for 'Resource' - 'Quantity', type '3'.

1. Again, copy *TwoSupport* and press Ctrl+Shift+V to paste, giving this new element the name *FiveSupport.*
2. Double-click on *FiveSupport* to open the 'Configure BPSim' dialog.
3. In the diagram, click on the 'Support' Resource element.
4. In the Configure BPSim window, in the 'Values' field for 'Resource' - 'Quantity', type '5'.

Run the simulations and do a comparison; in the Browser window:

1. Ctrl+click on *TwoSupport*, *ThreeSupport* and *FiveSupport*, then right-click and select the 'Show BPSim Configuration' option.
2. Ctrl+click on *TwoSupport-Result*, *ThreeSupport-Result* and *FiveSupport-Result*, then right-click and select the 'Show BPSim Report' option.



Tips:

- Click on the button and on the 'Show Only Different Items' option for both views
- You can dock the views together, so they provide direct comparisons: THESE are the differences in results caused by THOSE differences in configuration
- Toggle the filter bar to filter for the items that interest you

Analysis

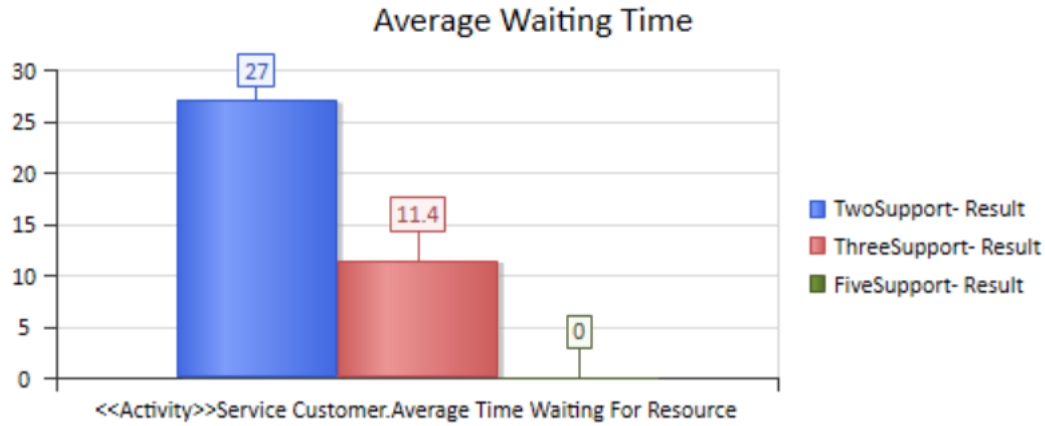
The docked comparison views show the configuration differences and the corresponding result differences.

- The customer's waiting time dropped from 27 minutes (2 Support resources) to 11.4 minutes (3 Support resources) and further down to 0 minutes (5 Support resources)
- The 'Degree Of Utilisation' dropped from 98% (2 Support resources) to 93% (3 Support resources) and further down to 83% (5 Support resources)

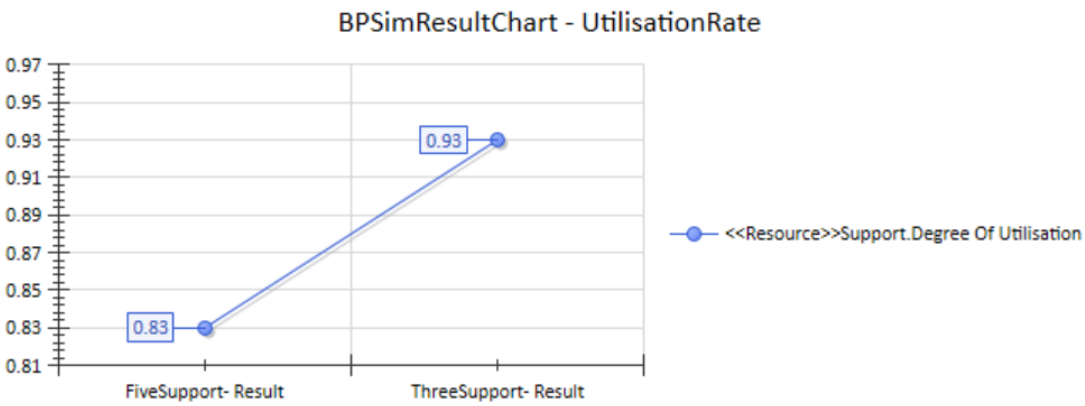
The customers will most likely be satisfied with 5 Support resources; however, the cost might be out of budget. So 3 or possibly 4 Support resources might be a balance point for this case. Try copying one of the Business Process Simulation Artifacts and configure and run a simulation for 4 Support resources.

Show Result with Chart

1. Drag a 'BPSim Result Chart' icon from the Toolbox onto the diagram and create a BPSim Result Chart Artifact; call it *Average Waiting Time.*
2. Right-click on the Artifact and select the 'Properties' option to display the element 'Properties' dialog; click on the 'BPSim Chart' page.
3. Click on the ... button and select a Base Report from which to define the schema (legends) to use in the chart; select *TwoSupport-Result.*
4. Choose the schema 'Time' | 'Service Customer' | 'Average Time Waiting For Resource'.
5. Click on the Add button to add another two BPSim Reports: *ThreeSupport-Result* and *FiveSupport-Result*
6. Click on the OK button and adjust the size of the Chart element; this Chart gave us very straightforward information



1. Create another BPSim Result Chart Artifact on the diagram, called *Utilisation Rate.*
2. Double-click on the Artifact to display the element 'Properties' dialog and click on the 'BPSim Chart' tab.
3. Click on the ... button and select a Base Report from which to define the schema (legends) to use in the Chart; select *TwoSupport-Result.*
4. Choose the schema 'Resource' | 'Support' | 'Degree Of Utilisation'.
5. Click on the Add button to add another two BPSim Reports: *ThreeSupport-Result* and *FiveSupport-Result.*
6. Click on the OK button and adjust the size of the Chart element.



Calendar-Based Help Desk Phone Support Simulation

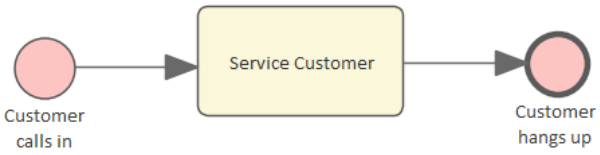
In this example, we create a very simple model to simulate a Help Desk telephone support process, based on calendar settings. We assume that:

- Customers call in at different intervals on weekdays and weekends
- Processing times differ between weekdays and weekends
- There are different numbers of support resources on weekdays and weekends

We model this process step by step, then create calendars and configure the Business Process simulation, which is simple enough to be calculated with pen and paper. After that, we run the simulation to compare that result with the pen and paper analysis.

Create BPMN Model

The model itself is very simple, consisting of a Start Event, a Task and an End Event.

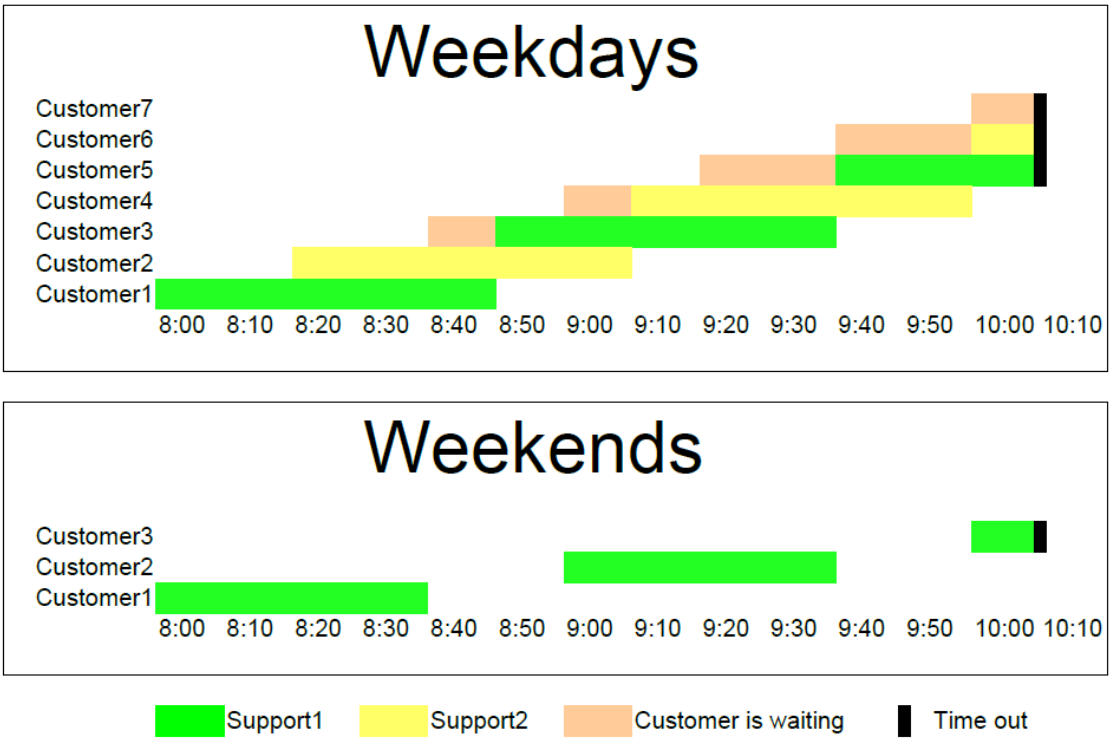


1. Create a Start Event *Customer calls in.*
2. Add a Sequence Flow to the target abstract task Activity *Service Customer.*
3. Add a Sequence Flow to the target End Event *Customer hangs up.*
4. Create a BPMN2.0 Resource named *Support.*
5. Create a BPMN2.0::ResourceRole inside *Service Customer*, give it the name *support* and set the tag *resourceRef* to the name of the Resource element *Support*.

Pen and Paper Analysis

We can use pen and paper to analyze this case:

- The simulation duration is 2 hours and 10 minutes, from 8:00AM to 10:10AM
- A customer calls in every 20 minutes on weekdays
- A customer calls in every 60 minutes on weekends
- To service each customer takes 50 minutes on weekdays
- To service each customer takes 40 minutes on weekends
- There are 2 support resources on weekdays
- There is 1 support resource on weekends



Looking at this result, when resource constraints are applied the computation is quite complicated for such a simple model.

On weekdays

- 7 customers called at intervals of 20 minutes over a duration of 2 hours and 10 minutes
- 4 customer calls were terminated normally
- 2 customer calls were interrupted because of timeout
- 1 customer call was not answered
- Support1 worked continuously for 130 minutes, Support2 worked continuously for 110 minutes

On weekends

- 3 customers called at intervals of 60 minutes over a duration of 2 hours and 10 minutes
- 2 customer calls were terminated normally
- 1 customer call was interrupted because of timeout
- Support1 worked 90 minutes, in 40-minute blocks with an interval of 20 minutes between calls

Now we will see how BPSim can help.

BPSim Configuration

In this section, we first create the Calendars, then we set up the Duration and Start parameters.

For element parameters, you can specify one or more calendars for a given parameter. However, **if any calendar is set for a parameter value, a default value (without any calendar specified) must exist**, otherwise the simulation will not work.

Clicking on the button on the Configure BPSim window toolbar will automatically check this constraint for you.

Task	Action
Create BPSim Artifact and Set Package	1. Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager'). 2. Create a Business Process Simulation Artifact named *Calendar Based Support Process Simulation.* 3. Select the Package containing the corresponding BPMN 2.0 model. 4. Open the diagram containing the model to be simulated.
Calendars	1. On the 'Configure' tab of the Configure BPSim window, click on the icon in the toolbar. The 'Edit BPSim Calendars' dialog displays. 2. Click on the New button to display the 'Event Recurrence' dialog and complete the fields as described here, to create a calendar. (You will create two calendars.) 3. In the 'Event Time' panel, set 'Start' to 08:00AM and 'End' to 05:00PM. 4. In the 'Recurrence Pattern' panel select 'Weekly' and select the checkboxes against 'Monday' through to 'Friday'. 5. In the 'Range of recurrence' panel set 'Start' to '02/11/2020' and select 'No end date'. 6. Click on the OK button. You are prompted to enter a Calendar name; overtype *Calendar_1* with 'Weekdays' and click on the OK button. 7. Click on the New button again and repeat steps 3 to 6 with these values: - 'Start' - 08:00AM - 'End' - 05:00PM - 'Weekly' - 'Saturday' and 'Sunday' - 'Start' to '07/11/2020' and 'No end date' - Overtype *Calendar_2* with 'Weekend' 8. Click on the OK button.
Duration	On the diagram, click on the *Calendar Based Support Process Simulation* BPSim Artifact and, on the 'Configure' tab of the Configure BPSim window, with the 'Calendar' field set to '----None----', create or edit this ScenarioParameter: • Duration - with a Constant value of 0000 002:10:00, which means 0 days, 2 hours and 10 minutes
Customer arrivals	On the diagram, click on the *Customer calls in* StartEvent and, on the 'Configure' tab of the Configure BPSim window, create or edit this Control parameter: • InterTriggerTimer - Value: 0 00:00:00, with the 'Calendar' field set to '----None----' (this default value is necessary) • InterTriggerTimer - Value: 0 00:20:00, with the 'Calendar' field set to 'Weekdays' • InterTriggerTimer - Value: 0 01:00:00, with the 'Calendar' field set to 'Weekends'
Processing Times	On the diagram, click on the *Service Customer* Activity and, on the 'Configure' tab of the Configure BPSim window, create or edit this Time parameter: • ProcessingTime - Value: 0 00:00:00, with the 'Calendar' field set to '----None----' (this default value is necessary) • ProcessingTime - Value: 0 00:50:00, with the 'Calendar' field set to 'Weekdays'

	• ProcessingTime - Value: 0 00:40:00, with the 'Calendar' field set to 'Weekends'
Resources	On the diagram, click on the *Support* Resource and, on the 'Configure' tab of the Configure BPSim window, create or edit this Resource parameter • Quantity - Value: 0; Calendar, with the 'Calendar' field set to '----None----' (this default value is necessary) • Quantity - Value: 2; Calendar, with the 'Calendar' field set to 'Weekdays' • Quantity - Value: 1; Calendar, with the 'Calendar' field set to 'Weekends'
Resource Selection (allocation)	On the diagram, click on the *Service Customer* Activity and, on the 'Configure' tab of the Configure BPSim window, with the Calendar field set to '----None----' check that the 'Values' field for the Resource parameter 'Selection' is set to: **bpsim::getResource('Support',1)** as an expression This expression is loaded from your BPMN model by default. You can do some advanced configurations for resource selection for a task.

Run Simulation

Weekdays

1. Click on the 'Calendar' field and select 'Weekday'.
2. Click on the 'Execute' tab and on the toolbar icon.

A file with the name *Calendar Based Support Process Simulation - Result* is generated. This report file contains the result for a weekday simulation, which is displayed on the 'Review' tab of the Configure BPSim window, on the 'Standard Results Report' tab.

Weekends

1. Click on the 'Calendar' field and select 'Weekend'.
2. Click on the 'Execute' tab and on the toolbar icon.

The *Calendar Based Support Process Simulation - Result* file is updated to show the result for a weekend simulation and displayed on the 'Review' tab of the Configure BPSim window, on the 'Standard Results Report' tab.

In each case, check the match between the result file and our analysis with pen and paper.

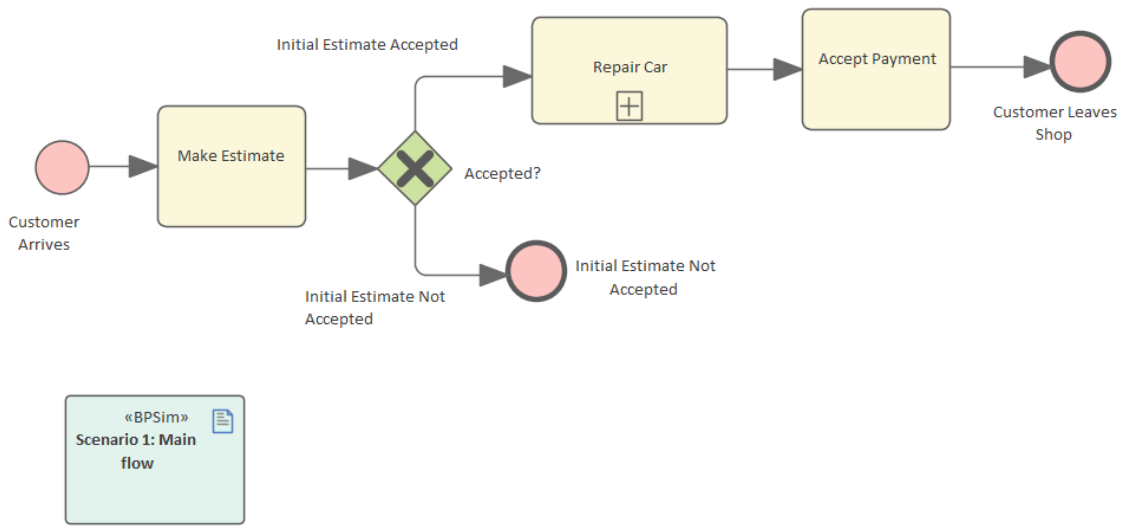
Car Repair Process

This example simulates the process flow of a Car Repair shop. The BPSim configuration:

- Uses a property parameter initialized by distribution to generate a random number of issues for each customer
- Applies probability to simulate:
  - Acceptance of the initial estimate or not
  - Whether new issues will be found during repair
- Increments or decrements the property parameter's value in each task
- Uses the property parameter's value on conditions of Sequences outgoing from Gateways
- Simulates the customer arrivals for a given start and duration

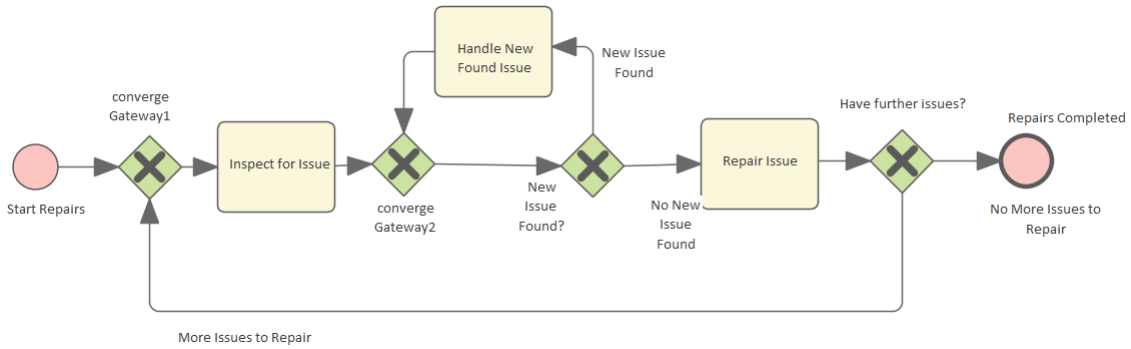
Create BPMN Model

Create the main process



1. Create a Start Event *Customer Arrives.*
2. Add a Sequence Flow to a target abstract task Activity *Make Estimate.*
3. Add a Sequence Flow to a target Exclusive Gateway *Accepted?.*
4. Add Sequence Flows to:
   - A target End Event *Initial Estimate Not Accepted*
   - A target subProcess *Repair Car*
5. From *Repair Car*, add a Sequence Flow to a target abstract task Activity *Accept Payment.*
6. Add a Sequence Flow to a target End Event *Customer Leaves Shop.*

Create the sub process Repair Car



1. Create a Start Event *Start Repairs.*
2. Add a Sequence Flow to a target Exclusive Gateway *converge Gateway1.*
3. Add a Sequence Flow to an abstract task Activity *Inspect for Issue.*
4. Add a Sequence Flow to an Exclusive Gateway *converge Gateway2.*
5. Add a Sequence Flow to an Exclusive Gateway *New Issue Found?.*
6. Add Sequence Flows to:
   - A target abstract task Activity *Handle New Found Issue*, then add a Sequence Flow back to *converge Gateway2*
   - A target abstract task *Repair Issue*, then add a Sequence Flow to a target Exclusive Gateway *Have further issues?*
7. From the Gateway *Have further issues?* add Sequence Flows to:
   - The target End Event *Repairs Completed*
   - *converge Gateway1*

Configure BPSim

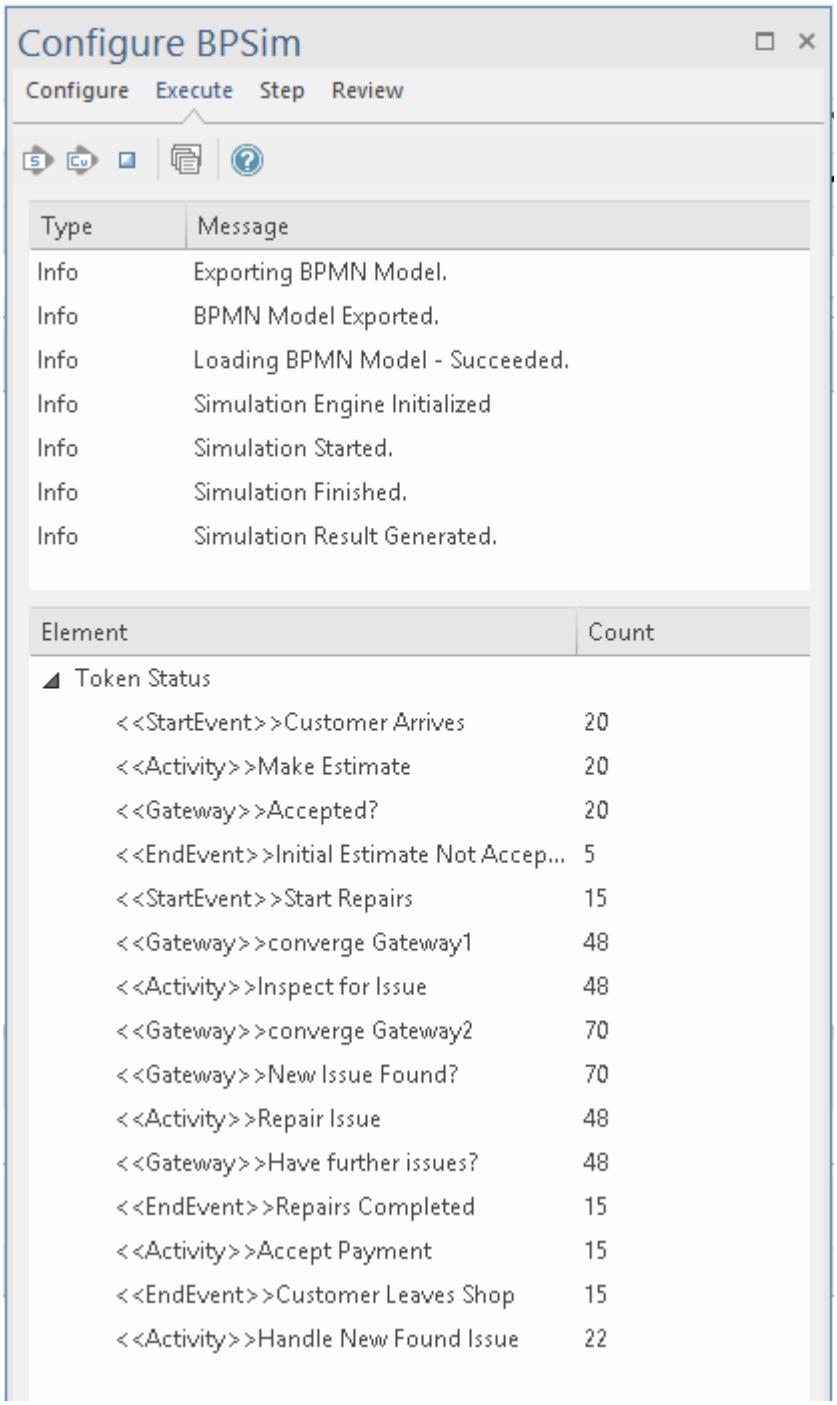
Task	Description
Artifact and Package	1. Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager'). 2. Create a Business Process Simulation Artifact named *Scenario 1: Main flow.* 3. Select the Package containing the corresponding BPMN 2.0 model.
Start and Duration	We will simulate the processes in a car repair shop whose opening hours are from 9:00 am to 5:00 pm, which is a period of 8 hours. We also suppose that a customer walking in after 4:50 pm will not be served on that day. Therefore the simulation Start time is 9:00 am and the duration is 7 hours and 50 minutes. On the 'Car Repair' diagram, click on the Business Process Simulation Artifact named *Scenario 1: Main Flow* and, on the Configure BPSim window, update these ScenarioParameters: • Start - overtype the 'Values' field with any date (in the format dd/mm/yyyy) and change the time section to '9:00 AM' • Duration - click on the ... button in the 'Values' field and set it to a Constant Duration of '0 07:50:00'
Customer Arrives	We will simulate a customer arriving every 24 minutes. The first customer arrives at 9:00 AM and the last arrives at 4:36 PM (the customer

	arriving at 5:00 PM will not be served today because that is constrained by the 'Duration' setting). With a pen and paper, we can calculate that there are 20 customers served (9:00 AM to 4:36 PM = 456 minutes; number of customers is 456/24 + 1 = 19 + 1 = 20). We will verify this with the simulation result later. On the 'Car Repair' diagram, click on the Start Event element *Customer Arrives*, and in the Configure BPSim window: 1. Click on the *New Parameter* drop-down arrow, and select 'Control'. 2. Click on the 'Parameter' drop-down arrow and select 'InterTriggerTimer'. 3. In the 'Values' field click on the button and set a Constant Numeric value of '24 minutes'. Click on the OK button and the Save toolbar icon.
Property Parameters	We suppose each customer's car might initially have a different number of issues. This could be reflected using a random number generator. BPSim provides a number of distributions to suit your needs. In this example, we use a Truncated Normal distribution to initialize the property *noOfIssues.* Tasks *Repair Issue* and *Handle New Found Issue* will decrement and increment the value of *noOfIssues* respectively. 1. On the 'Car Repair' diagram, click on the Start Event *Customer Arrives*. 2. On the 'Configure' tab of the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Property parameter called *noOfIssues.* 3. In the 'Values' field click on the button; the 'Configure 'noOfIssues' for 'CustomerArrives" dialog displays. 4. Click on the 'Distribution' tab and select 'TruncatedNormal'; in the fields: - 'Mean', type '2' - 'StandardDeviation', type '1' - 'Min', type '1' - 'Max', type '1000' **Important Note:** Distributions such as 'TruncatedNormal', return a floating point value, but the property is used as an integer. Setting the property's type is important, especially in condition expressions when testing with equality. For example, the condition expression *getProperty('noOfIssues') = 0* will almost never be satisfied because *noOfIssues* was initialized by a floating point distribution. **Tip: How to customize the type for a property** After you create the property and set a value, click on the icon on the toolbar, and then click on the icon to display the 'Edit Property Parameters' dialog. In the 'Type' field for the property, click on the drop-down arrow and select the value 'int' instead of the default 'double'. 1. On the 'Repair Car' diagram, click on the Activity *Repair Issue.* 2. On the 'Configure' tab of the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Property parameter called *noOfIssues.* 3. In the 'Values' field click on the button. The 'Configure 'noOfIssues' for 'Repair Issue" dialog displays. 4. Click on the 'Expression' tab and, in the 'Expression' field, type *{noOfIssues} -1*; click on the OK button. 5. On the 'Repair Car' diagram, click on the Activity *Handle New Found Issue*. 6. On the 'Configure' tab of the Configure BPSim window, click on the *New*

	Parameter drop-down arrow and create a Property parameter called *noOfIssues.* 7. In the 'Values' field click on the ... button. The 'Configure 'noOfIssues' for 'Handle New Found Issue" dialog displays. 8. Click on the 'Expression' tab and, in the 'Expression' field, type *{noOfIssues} +1*; click on the OK button.
Probability on Sequence Flows	We estimate that one in three of the customers will not accept the initial estimate for repairs and the remaining two will accept it. We also estimate that for one in four of the repairs new issues will be found, and for the remaining three repairs no new issues will be found. On the 'Car Repair' diagram, refer to the Gateway element *Accepted?*. Click on the: • *Initial Estimate Accepted* Sequence Flow and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Probability'; in the 'Values' field type '0.67' • *Initial Estimate Not Accepted* Sequence Flow, and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Probability'; in the 'Values' field type '0.33' On the 'Car Repair' diagram, refer to the Gateway element *New Issue Found?*. Click on the: • *No More Issues to Repair* Sequence Flow and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Probability'; in the 'Values' field type '0.75' • *More Issues to Repair* Sequence Flow and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Probability'; in the 'Values' field type '0.25'
Condition on Sequence flows	We use an expression to return a Boolean value as a Sequence Flow's condition, which plays a key role in the flow's logic. On the 'Repair Car' diagram, refer to the *Have Further Issues?* Gateway element. Click on the: • *More Issues to Repair* Sequence Flow and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Condition'; in the 'Values' field click on the ... button, click on the 'Expression' tab and type **{noOfIssues} != 0** in the 'Expression' field • *No More Issues to Repair* Sequence Flow and in the Configure BPSim window click on the *New Parameter* drop-down arrow, and create a Control parameter called 'Condition'; in the 'Values' field click on the ... button, click on the 'Expression' tab and type **{noOfIssues} = 0** in the 'Expression' field Note: All the outgoing transitions from a Gateway should include 100% of the logic; for example, you would not enter **{noOfIssues} > 10** and **{noOfIssues} < 5** as condition expressions, because values in the range **[5, 10]** will not be handled by any outgoing Sequence Flows.

Run Simulation

1. On the Configure BPSim window click on the 'Execute' tab and on the icon in the toolbar.
2. When the simulation is complete, the Execute tab provides results similar to these:



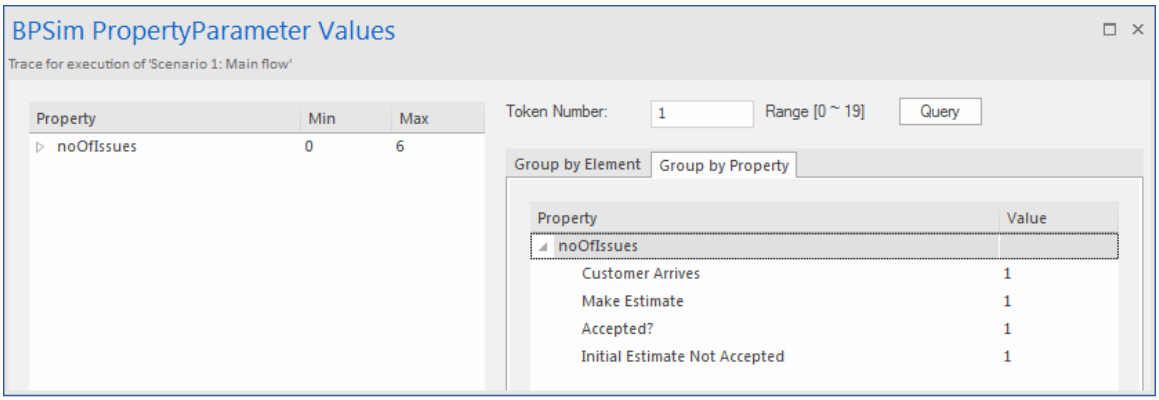
Token Analysis

- 20 customers arrived, matching the number we calculated manually (see *Customer Arrives* in the *Configure BPSim* table)
- 8 customers out of the 20 did not accept the initial estimate, whilst 12 of the 20 accepted and had their car repaired; these figures roughly match the 1/3 and 2/3 probabilities
- 64 tokens passed the Gateway *New Issue Found?*, of which 19 had new issues and 45 did not; these figures roughly match the 1/4 and 3/4 probabilities

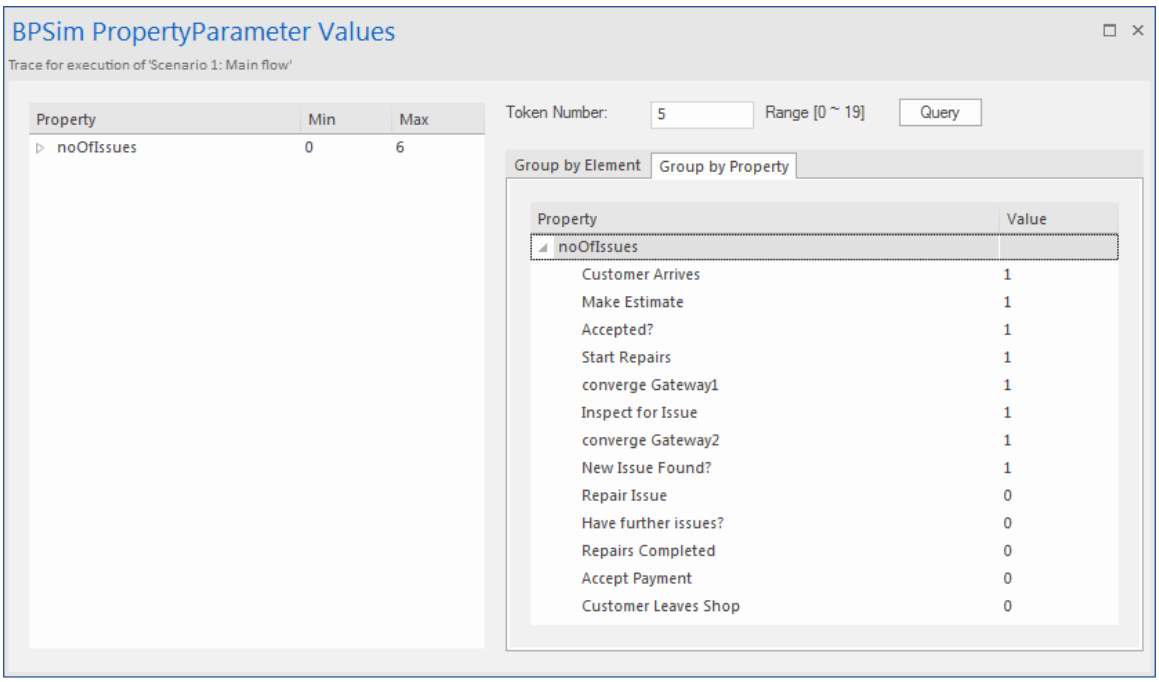
Analysis on individual customers

Click on the button on the toolbar to open the 'BPSim PropertyParameter Values' dialog. As there are 20 customers (tokens), you can input a value between 0 and 19 in the 'Token Number' field and click on the Query button to do some analysis:

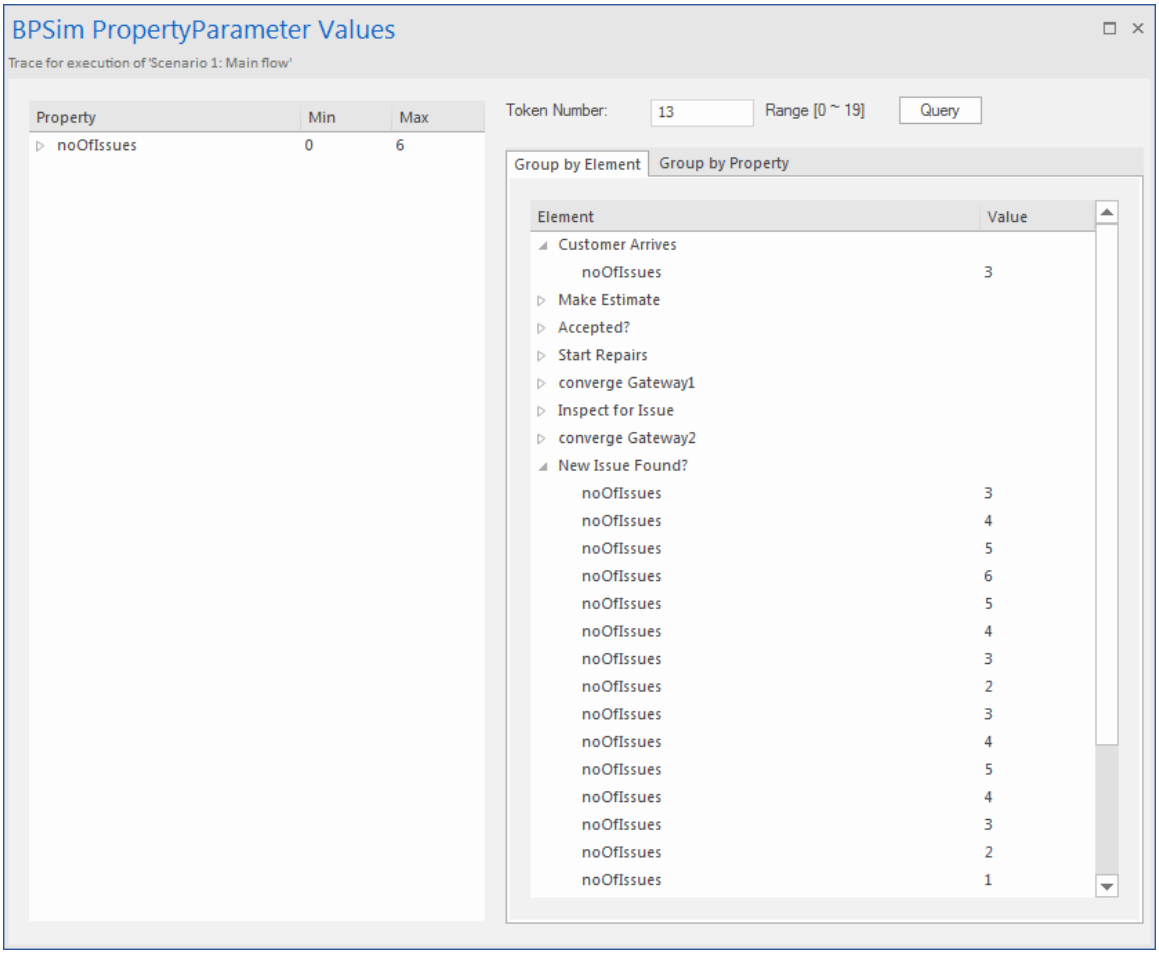
- This customer did not accept the initial estimate, as shown on the 'Group by Property' tab:



- This customer's car had only one issue, which was fixed:



- This customer's car had three known issues and three other issues were found during repair, so in total six issues got fixed (perhaps this is a really old car); switching to the 'Group by Element' tab:


BPSim PropertyParameter Values
Trace for execution of 'Scenario 1: Main flow'
Property
Min
Max
noOfIssues
0
6
Token Number:
13
Range [0 ~ 19]
Query
Group by Element
Group by Property
Element
Value
Customer Arrives
noOfIssues
3
Make Estimate
Accepted?
Start Repairs
converge Gateway1
Inspect for Issue
converge Gateway2
New Issue Found?
noOfIssues
3
noOfIssues
4
noOfIssues
5
noOfIssues
6
noOfIssues
5
noOfIssues
4
noOfIssues
3
noOfIssues
2
noOfIssues
3
noOfIssues
4
noOfIssues
5
noOfIssues
4
noOfIssues
3
noOfIssues
2
noOfIssues
1

BPMN2.0 Event Examples

An Event is something that happens during the course of a Process. Events affect the flow of the Process, usually having a cause or an impact, and in general requiring or allowing for a reaction. For example, the start of an Activity, the end of an Activity, the change of state of a document, or the arrival of a Message could all be considered to be Events.

Events allow for the description of 'event-driven' Processes. In these Processes, there are three main types of Event:

- Start Events, which indicate where a Process will start
- End Events, which indicate where the path of a Process will end
- Intermediate Events, which indicate where something happens between the start and end of a Process

Within these three types, Events can be one of two subtypes:

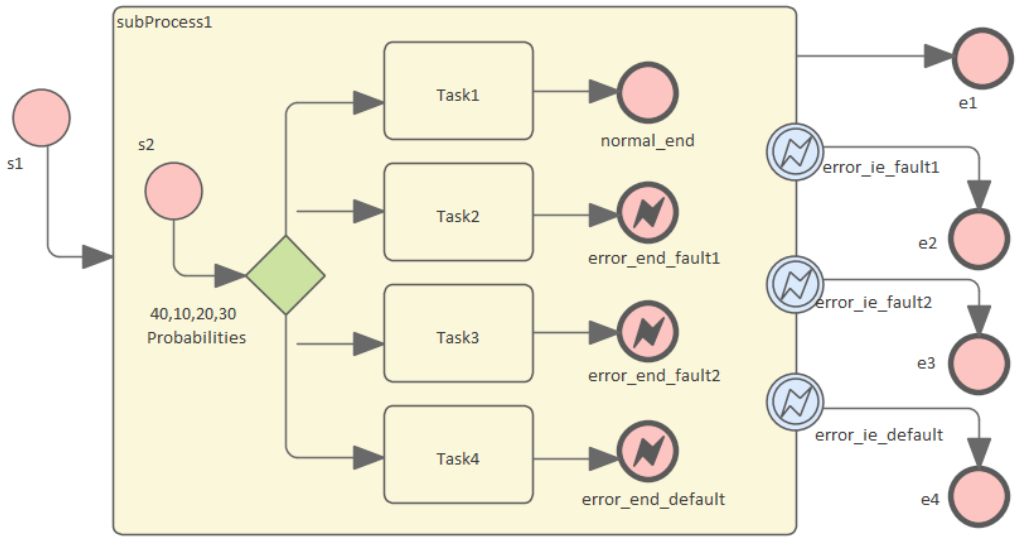
- Events that catch a trigger - all Start Events and some Intermediate Events are catching Events
- Events that throw a Result - all End Events and some Intermediate Events are throwing Events that could eventually be caught by another Event

In this section, we provide examples illustrating many of the commonly-used BPMN 2.0 events. In each example, we provide step-by-step BPMN modeling and BPSim configuration instructions, and a thorough analysis of the simulation result. All the examples are available in the EAExample model.

Error Event

When an Intermediate Error Event connects to the boundary of an Activity, it becomes part of an exception flow. The event is triggered when a token causes a fault name to be raised in the normal flow, going through to an Error End Event.

Create BPMN Model



Create the main process

- Create a Start Event *s1*
- Add a Sequence Flow to a target Activity element *subProcess1;* enlarge the Activity and right-click, selecting the 'Is Expanded' option, then open the 'Properties' dialog and set 'Type' to 'subprocess'
- Add a Sequence Flow to a target End Event element *e1* ('Type' set to 'None')
- Create three Boundary Intermediate Events, dragging the elements from the Toolbox and dropping them on *subProcess1*; from the instant menus select 'Edge-Mounted' and 'Error':
  - *error_ie_fault1*; add a Sequence Flow to a target EndEvent element *e2* ('Type' set to 'None')
  - *error_ie_fault2*; add a Sequence Flow to a target EndEvent element *e3* ('Type' set to 'None')
  - *error_ie_default*; add a Sequence Flow to a target EndEvent elemenr *e4* ('Type' set to 'None')

Create the subprocess

Within the *subProcess1* Activity:

- Create a Start Event *s2*, 'Standalone' and set 'Type' to 'None'
- Create a Sequence Flow to a target Gateway element set to 'Exclusive' and with the name '*40,10,20,30 Probabilities'*
- Create Sequence Flows to four target Activiy elements of Type 'abstractTask' called:
  - *Task1*, and add a Sequence Flow to a target EndEvent called *normal_end*, 'Type' set to 'None'
  - *Task2*, and add a Sequence Flow to a target EndEvent called *error_end_fault1,* 'Type' set to 'Error'
  - *Task3*, and add a Sequence Flow to a target EndEvent called *error_end_fault2,* 'Type' set to 'Error'
  - *Task4*, and add a Sequence Flow to a target EndEvent called *error_end_default,* 'Type' set to 'Error'

Create BPMN2.0::Error elements

Create the Error elements *Fault1* and *Fault2*, which will be used as error code by Events.

- Double-click on the *error_end_fault1* element and, in the 'Properties' dialog 'BPMN2.0' tab, locate the 'errorRef' tag
- In the 'Value' field, click on the ... button and browse to the Package containing this model
- Click on the Add New button and, in the 'Name' field, type the name *Fault1*, then click on the Save button
- Again click on the Add New button and, in the 'Name' field, type the name *Fault2*, then click on the Save button
- Click on the OK button, and again on the next OK button

Set up Events for Error Codes

- Double-click on the *error_end_fault1* element and, in the 'Properties' dialog 'BPMN2.0' tab, locate the 'errorRef' tag
- In the 'Value' field, click on the ... button and browse to the Package containing this model
- Click on *Fault1*, then on the OK button, and again on the OK button.

Do the same for these elements:

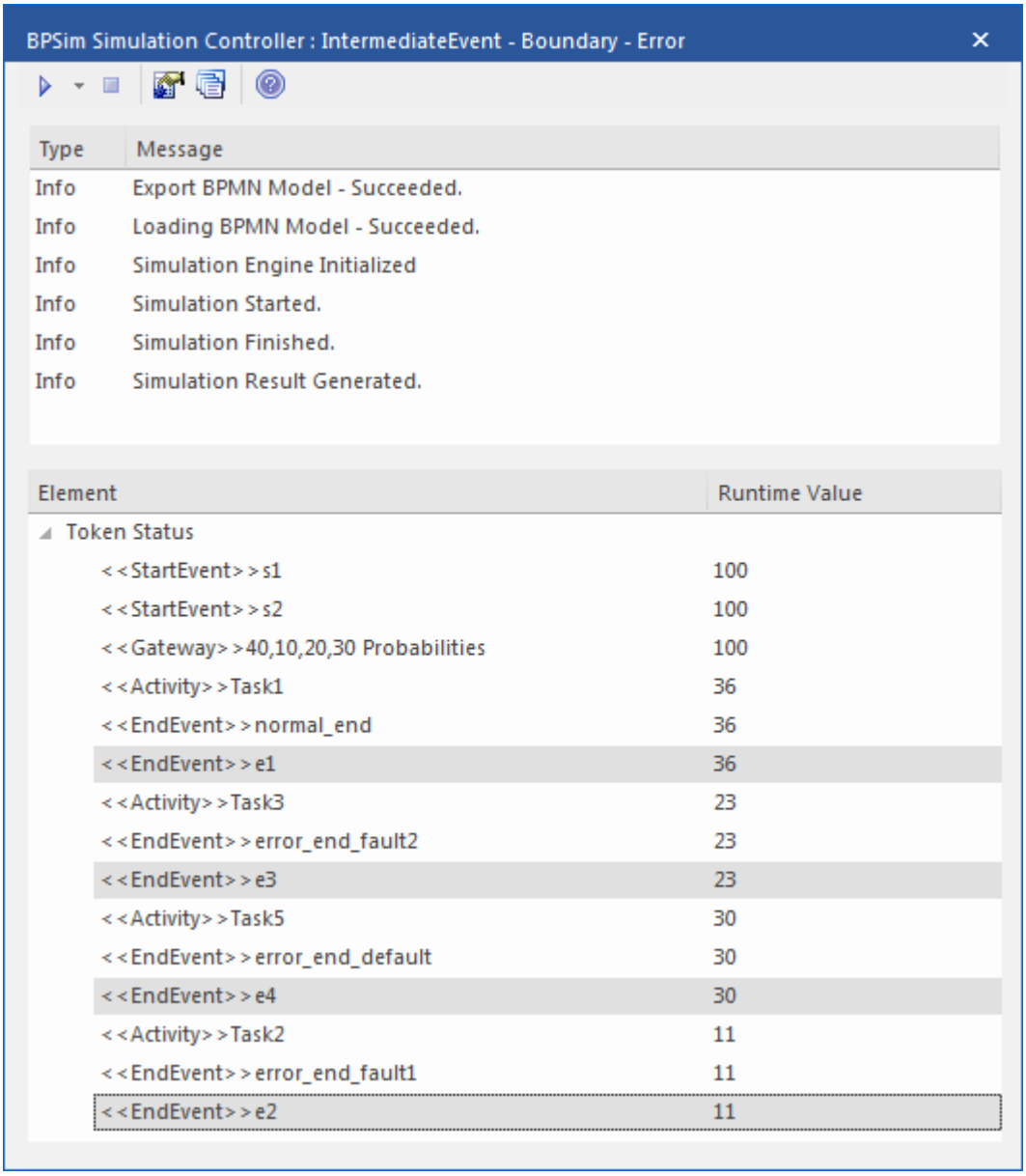
- *error_end_fault2*, clicking on *Fault2*
- *error_ie_fault1*, clicking on *Fault1*
- *error_ie_fault2*, clicking on *Fault2*

Configure BPSim

Object	Action
Artifact & Package	• Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'IntermediateEvent - Boundary - Error' (in the 'Select/Create Artifact' field, click on the ... button and select its parent Package and click on the Add New button, then type in the element name and click on the Save button and the OK button) Then all the BPMN elements will be loaded in to the Configure BPSim window.
s1	• From the tree on the left of the Configure BPSim window, expand 'StartEvent' and click on 's1' • On the 'Control' tab, in the 'New Parameter...' field, click on the drop-down arrow and select 'TriggerCount' • In the 'Value' field, type '100'
Probability	From the tree on the left of the Configure BPSim window, expand 'Gateway | 40,10,20,30 Probabilities'. *Tips: You can also float the Configure BPSim window, then click on the element or connectors on the BPMN diagram; the element in the Configure BPSim window will be automatically selected.* For each of the *Taskn* elements, in the 'Control' tab click on the 'New Parameter' drop-down arrow and select 'Probability', then type the corresponding value in the 'Value' field: • For Task1 type '0.4' • For Task2 type '0.1' • For Task3 type '0.2' • For Task4 type '0.3'

Run Simulation

- On the 'Configure BPSim' dialog Toolbar, click on the 'Run' icon to open the 'BPSim Controller' dialog
- Click on the Run button and select 'Standard Simulation'
- The results of the simulation resemble this:



Analysis:

The Probability set on the Sequence Flows outgoing from *40,10,20,30 Probabilities* are 0.4, 0.1, 0.2 and 0.3 respectively.

- 36 out of 100 passes finished at *normal_end*, which flowed to *e1*
- 11 out of 100 passes finished at *error_end_fault1*, which triggered *error_ie_fault1* by the ErrorRef *Fault1,* and the exception flowed to *e2*
- 23 out of 100 passes finished at *error_end_fault2*, which triggered *error_ie_fault2* by ErrorRef *Fault2,* and the

exception flowed to *e3*

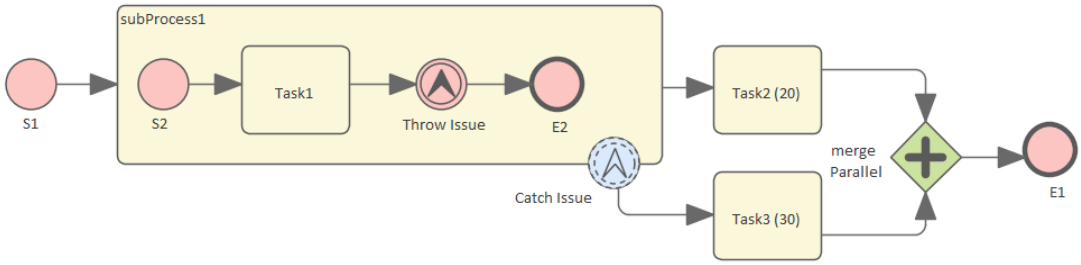
- 30 out of 100 passes finished at *error_end_default*, which triggered *error_ie_default* because they did not set ErrorRef and the exception flowed to *e4*

The numbers 36, 11, 23 and 30 add up to 100, which was set as the TriggerCount in *s1,* so they match the 100% probability

Escalation Event

In BPMN, Escalation is the non-interrupting counterpart of Error, with similar throw-catch behavior. Unlike Error, however, the normal flow and exception flow exits from the Activity are parallel paths, not alternative.

Create BPMN Model



Create the main process

- Create a Start Event *S1*
- Add a Sequence Flow to a target Activity *subProcess1;* enlarge the Activity and right-click, selecting the 'Is Expanded' option, then open the 'Properties' dialog and set 'Type' to 'subprocess'
- Add a Sequence Flow to a target abstractTask Activity element *Task2 (20)* (open the 'Properties' dialog and set the 'Type' field to 'abstractTask')
- Add a Sequence Flow to a target parallel Gateway element *merge Parallel* (open the 'Properties' dialog and set the 'Type' field to 'parallel')
- Add a Sequence Flow to a target End Event *E1*
- On *subProcess1*, add a boundary non-interrupting Escalation Event *Catch Issue* (drag the 'Intermediate Event' icon onto *subProcess1*, and from the instant menus select 'Edge Mounted' and 'Escalation'; double-click on the element to display the 'Properties' dialog and add the name, then in the 'Type' field select 'Boundary Non-Interrupting > Escallation')
- Add a Sequence Flow to a target abstractTask Activity element *Task3 (30)* (open the 'Properties' dialog and set the 'Type' field to 'abstractTask')
- Add a Sequence Flow to the target element *merge Parallel*

Create the sub process

- Within (or under) *subProcess1*, create a Start Event *S2*
- Add a Sequence Flow to a target abstractTask Activity element *Task1* (open the 'Properties' dialog and set the 'Type' field to 'abstractTask')
- Add a Sequence Flow to a target Throwing Escalation Intermediate Event *Throw Issue* (open the 'Properties' dialog and in the 'Type' field select 'Throwing > Escalation')
- Add a Sequence Flow to a target End Event *E2*

Create BPMN2.0::Escalation elements

From the Diagram Toolbox, expand the 'BPMN 2.0 Types' page, drag the 'Escalation' icon onto the diagram, and give the element the name *Escalation1*; this will be used as the escalation code by the Events.

Set up Events for Escalation Codes:

- Double-click on *Throw Issue* and in the 'Value' field for the escalationRef tag click on the icon and locate and select *Escalation1*
- Double-click on *Catch Issue* and, again, in the 'Value' field for the escalationRef tag click on the icon and locate and select *Escalation1*

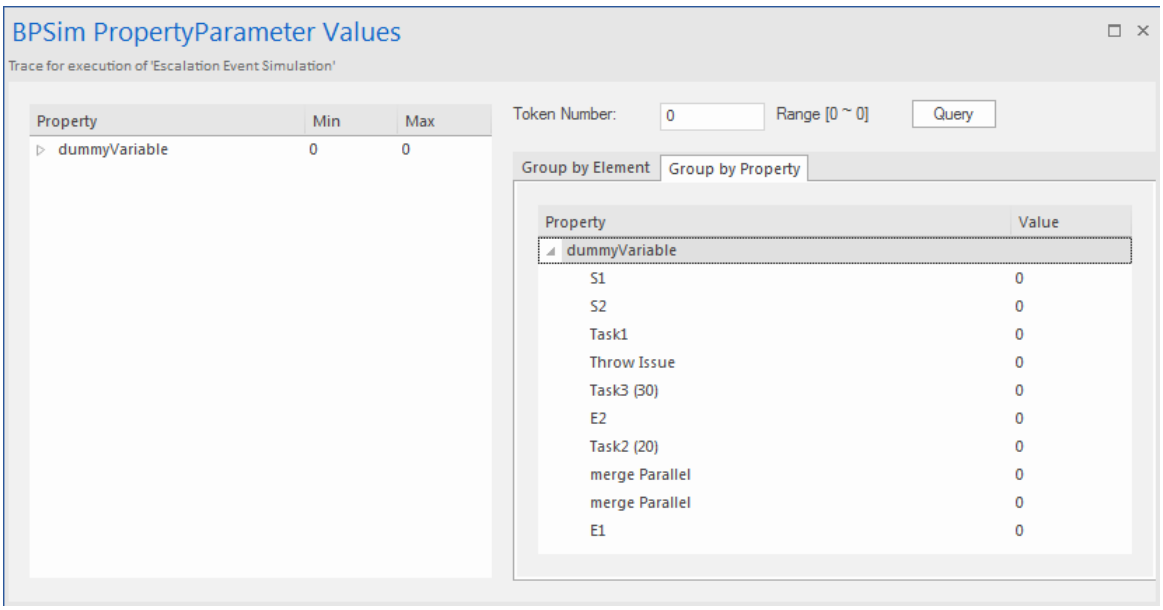
(The exception flow exits from the Activity are parallel.)

Configure BPSim

Task	Action
Artifact & Package	• Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'Escalation Event Simulation' (in the 'Select/Create Artifact' field, click on the button and select its parent Package and click on the Add New button, then type in the element name and click on the Save button and the OK button) Then all the BPMN elements will be loaded in to the Configure BPSim window.
Trigger Count of Start Event	• From the tree on the left of the Configure BPSim window, expand 'StartEvent' and click on *S1* • On the 'Control' tab, in the 'New Parameter...' field, click on the drop-down arrow and select 'TriggerCount' • In the 'Value' field, type '1'
ProcessingTime	• In the left-hand tree expand 'Activity' and click on *Task2 (20)*; in the 'Value' field for 'Processing Time' type '20' and in the 'Unit' field type 's' (for 20 seconds) • Click on *Task3 (30);* in the same way, set 'ProcessingTime' to 30 seconds
dummyVariable for Trace	In order to show the exact trace of a given token, you must set a dummy variable on *S1*. • In the left-hand hierarchy click on *S1*, and on the 'Properties' tab overtype the *New Property* text with the name of a variable (such as 'dummyVariable') • In the 'Value' field, click on the button and, on the '<<StartEvent>>S1 : <variable name>' dialog click on 'Numeric' and type a 'Constant Numeric' value of '0'; click on the OK button

Run Simulation

- On the 'Configure BPSim' dialog Toolbar, click on the 'Run' icon to open the 'BPSim Simulation Controller' dialog
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'
- After simulation, click on the button on the tool bar to display the 'BPSim PropertyParameter Values' dialog
- Click on the Query button and on the 'Group by Property' tab, and expand 'dummyVariable' (or the name you assigned to the variable)



Analysis:

Unlike *Error*, the normal flow and exception flow exits from *subProcess1* are not alternative paths but parallel. This feature can be easily discovered from the trace:

- *E2* and *Task2 (20)* still get traversed after*Task3 (30)* started
- *E1* was reached after *mergeParallel* was traversed twice

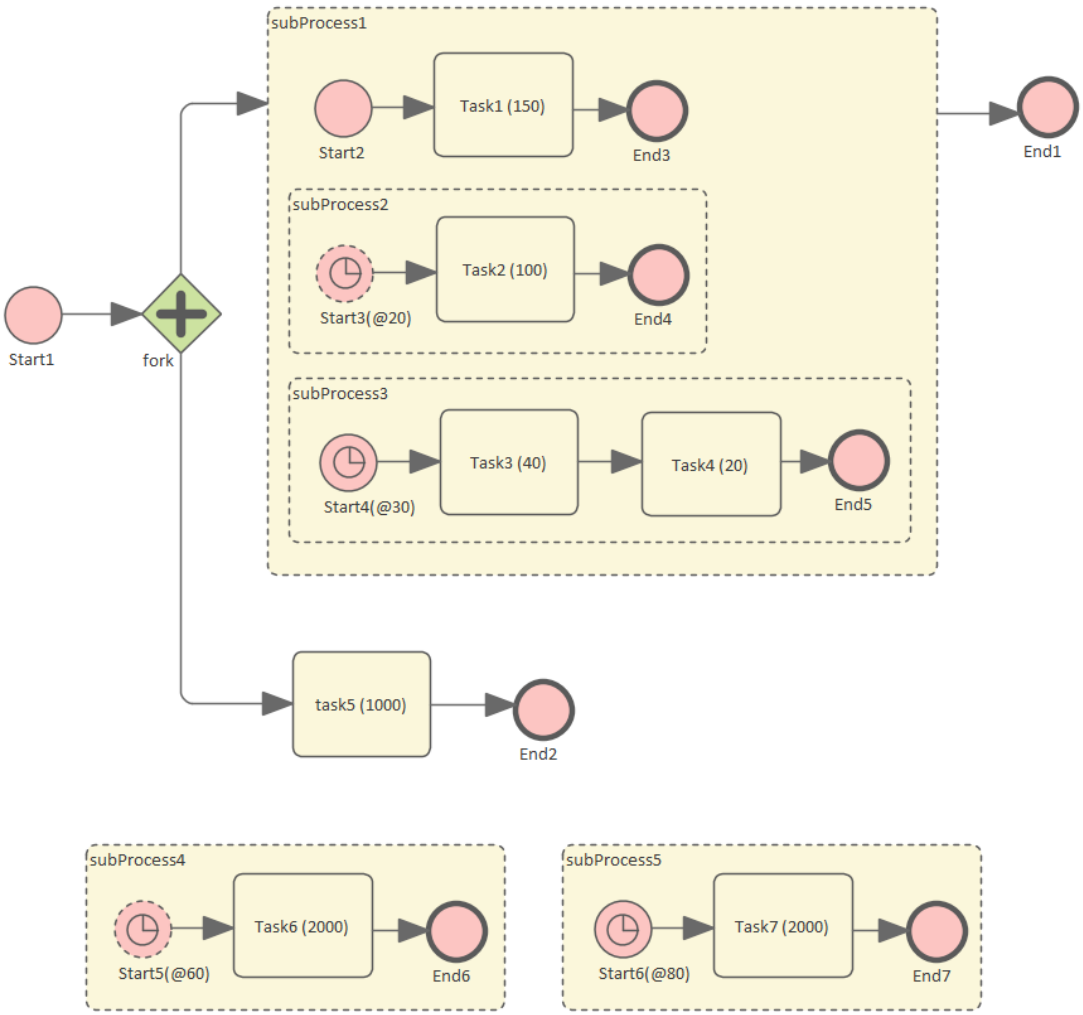
Event Sub-Process

Event Sub-Processes enable your system to handle an Event within the context of a given Sub-Process or Process. An Event Sub-Process always begins with a Start Event; it is not instantiated by normal control flow, but only when the associated Start Event is triggered. Event Sub-Processes are self-contained and MUST not be connected to the rest of the Sequence Flows in the Sub-Processes.

- If the isInterrupting attribute of its Start Event is set, an Event Sub-Process cancels execution of the enclosing Sub-Process
- If the isInterrupting attribute is not set, execution of the enclosing Sub-Process continues in parallel to the Event Sub-Process

In this example, we demonstrate how Interrupting and Non-Interrupting Event Sub-Processes affect the life line of the enclosing Sub-Process and Process.

Create BPMN Model



Model the main process

- Create a StartEvent *Start1*
- Add a Sequence Flow to a target Parallel Gateway element *fork*
- Add a Sequence Flow to
  - a Sub-Process *subProcess1*, and from that add a *S*equence Flow to a target End Event element

End1
- an abstract Task *Task5*, and from that add a Sequence Flow to a target End Event element *End2*

Tips on how to model an Event Sub-Process

- Drag an Activity from the 'BPMN2.0 - Business Process' toolbox onto the diagram
- Double-click on the Activity to display the 'Properties' dialog and, in the 'Type' field, select 'subProcess'; set 'triggeredByEvent' to 'true' and click on the OK button
- Right-click on the element and select the 'Is Expanded' option; this will display the element name on the top left corner

Model the Event Sub-Processes for the main process

- Create an Event Sub-Process *subProcess4*
  - Create a Timer Start Event *Start5(@60)*, then double-click on it to display the 'Properties' dialog and, in the 'Type' field, select 'Event Sub-Process Non-Interrupting > Timer'; click on the OK button
  - Add a Sequence Flow to a target abstract task Activity *Task6(2000)*
  - Add a Sequence Flow to a target End Event element *End6*
- Create an Event Sub-Process *subProcess5*
  - Create a Timer Start Event *Start6(@80)*, then double-click on it to display the 'Properties' dialog and, in the 'Type' field, select 'Event Sub-Process Interrupting > Timer'; click on the OK button
  - Add a Sequence Flow to a target abstract task Activity *Task7(2000)*
  - Add a Sequence Flow to a target End Event element *End7*

Model the Sub-Process subProcess1 and the enclosed Event Sub-Processes

- Create a StartEvent *Start2*
  - Add a *S*equence Flow to a target abstract task Activity *Task1(150)*
  - Add a Sequence Flow to a target End Event *End3*
- Create an Event Sub-Process *subProcess2*
  - Create a Timer Start Event *Start3(@20)*, then double-click on it to display the 'Properties' dialog and, in the 'Type' field, select 'Event Sub-Process Non-Interrupting > Timer'
  - Add a Sequence Flow to a target abstract task Activity *Task2(100)*
  - Add a Sequence Flow to a target End Event element *End4*
- Create an Event Sub-Process *subProcess3*
  - Create a Timer Start Event *Start4(@30)*, then double-click on it to display the 'Properties' dialog and, in the 'Type' field, select 'Event Sub-Process Interrupting > Timer'
  - Add a Sequence Flow to a target abstract task Activity *Task3(40)*
  - Add a Sequence Flow to a target abstract task Activity *Task4(20)*
  - Add a Sequence Flow to a target End Event element *End5*

Configure BPSim

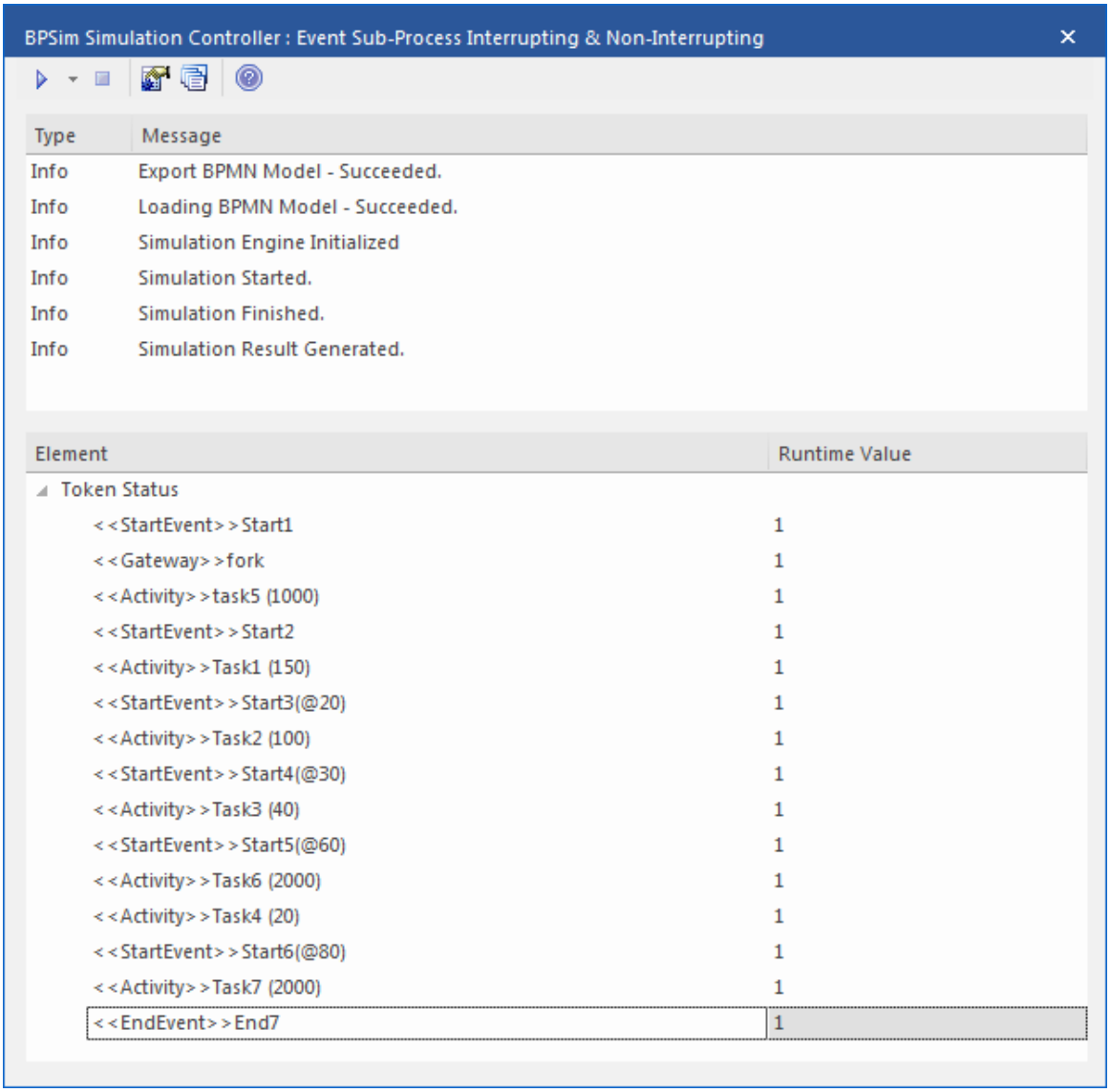
Using this table, we create the Artifact in the configuration Package and configure the parameter values of each element.

Task	Action
Create Artifact	• Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'Event Sub Process Interrupting and Non-Interrupting' (in the 'Select/Create Artifact' field, click on the button and select its parent Package and click on the Add New button, then type in the element name and click on the Save button and the OK button)

	Then all the BPMN elements will be loaded in to the Configure BPSim window.
InterTriggerTimer for Start Events in Event Sub-Process	From the tree on the left of the 'Configure BPSim' dialog, expand 'StartEvent'. For each of the elements listed here, on the 'Control' tab click on the drop-down arrow in the 'New Parameter...' field and select the parameter 'InterTriggerTimer'. Click on the button in the 'Value' field to open the 'Parameter' dialog and select 'Constant > Numeric', then type in the value and select 'seconds'. • Start3(@20): 20 seconds • Start4(@30): 30 seconds • Start5(@60): 60 seconds • Start6(@80): 80 seconds
ProcessingTime for Tasks	From the tree on the left of the Configure BPSim window, expand 'Activity'. For each of the elements listed here, on the 'Time' tab click on the drop-down arrow in the 'New Parameter...' field and select the parameter 'ProcessingTime'. Click on the button in the 'Value' field to open the 'Parameter' dialog and select 'Constant > Numeric', then type in the value and select 'seconds'. • Task1 (150): 150 seconds • Task2 (100): 100 seconds • Task3 (40): 40 seconds • Task4 (20): 20 seconds • Task5 (1000): 1000 seconds • Task6 (2000): 2000 seconds • Task7 (2000): 2000 seconds

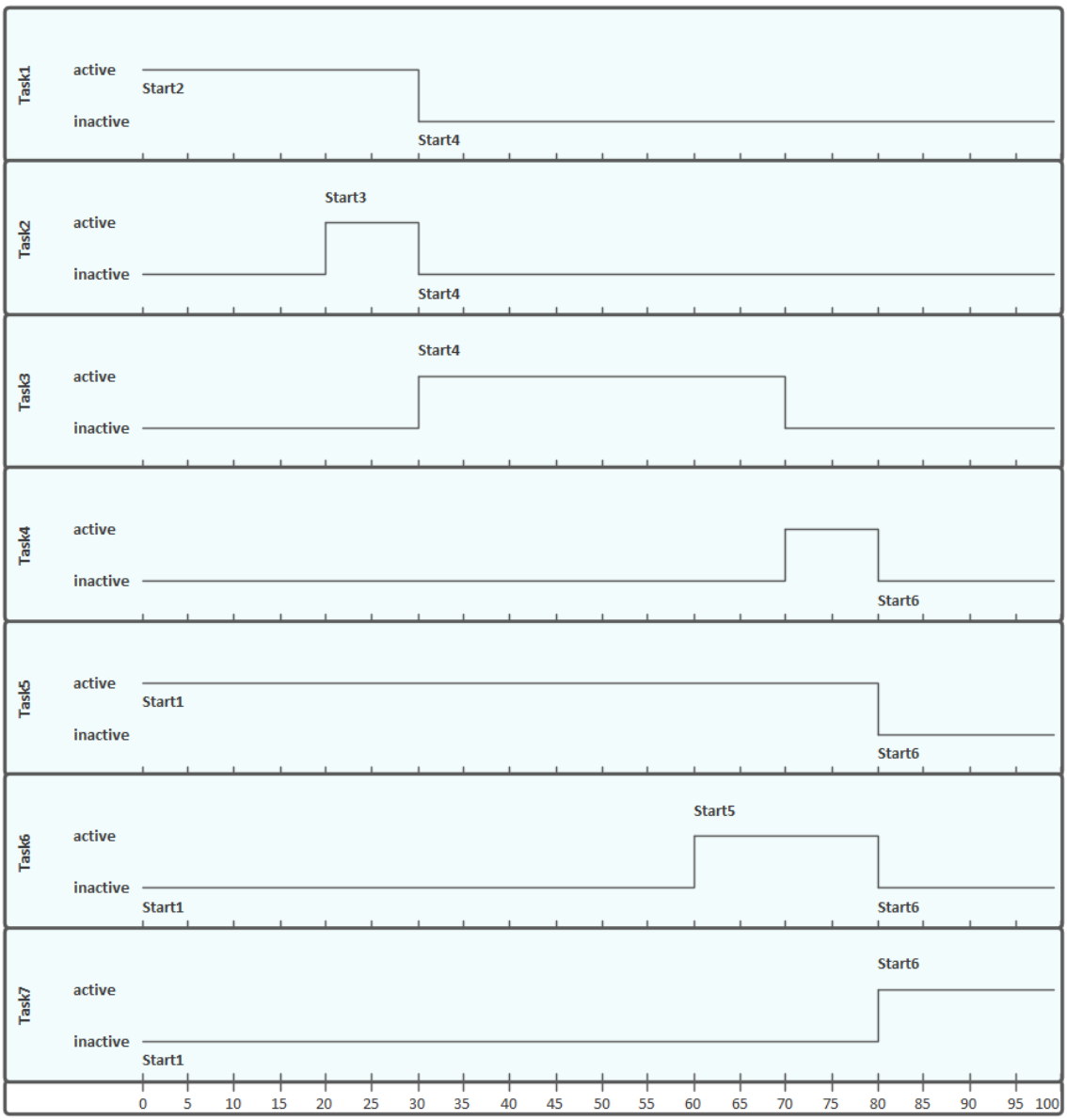
Run Simulation

- On the 'Configure BPSim' dialog Toolbar, click on the 'Run' icon to open the 'BPSim Simulation Controller' dialog
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'



Analysis

Reading the results, it might not be completely straightforward to see what has happened; however, if we draw the lifeline for each task in a Timing diagram, it becomes clearer.



- Event *Start3(@20)* is Non-Interrupting, it did not stop *Task1* at 20 seconds
- Event *Start4(@30)* is Interrupting, it stopped *Task1* and *Task2* at 30 seconds; it did not affected *Task5* because *Task5*'s enclosing process (main process) level is higher than *Start4*'s enclosing Sub-Process (*subProcess1*)
- Event *Start5(@60)* is Non-Interrupting, it started *Task6* at 60 seconds without affecting *Task3* or *Task5*
- Event *Start6(@80)* is Interrupting, it started *Task7* at 80 seconds and interrupted the running Tasks (*Task4*, *Task5*, *Task6*) that were in the same or lower level of its enclosing process
- Only *End7* is reached as expected

Fibonacci Number Generator with Link Event

A Link Event is a mechanism for connecting two sections of a Process. Link Events can be used:

- To create looping scenarios, as generic 'Go To' objects within the Process level
- To avoid long Sequence Flow lines; paired Link Events can be used as 'off-page' connectors for printing a Process across multiple pages

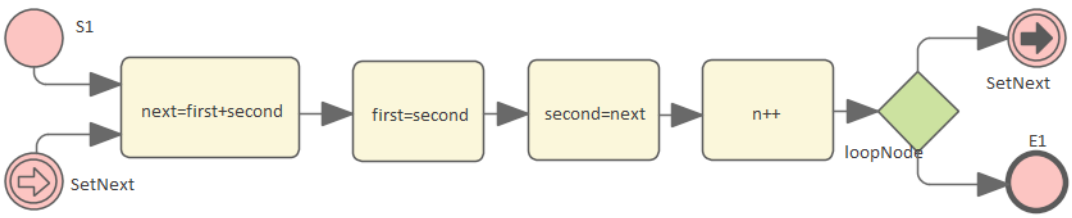
The use of Link Events is limited to a single Process level (that is, they cannot link a parent Process with a sub-Process).

There can be multiple source Link Events, but there can only be one target Link Event.

- The target Link Event marker is unfilled, to 'catch' from the source link
- The source Link Event marker is filled to 'throw' to the target link

When the EABPSim Execution Engine is running the simulation, the source-target Link Events are paired by element NAME, so they cannot be empty.

Create BPMN Model



- Create a StartEvent *S1*
- Add a Sequence Flow to a target abstractTask Activity element *next=first+second* (open the 'Properties' dialog and set the 'Type' field to 'abstractTask')
- Add a Sequence Flow to a target abstractTask Activity element *first=second*
- Add a Sequence Flow to a target abstractTask Activity element *second=next*
- Add a Sequence Flow to a target abstractTask Activity element *n++*
- Add a Sequence Flow to a target exclusive Gateway element *loopNode* (on the instant menu, select 'Exclusive')
- Add a Sequence Flow to each of these target elements:
  - A Throwing Link Intermediate Event element *SetNext* (open the 'Properties' dialog and set the 'Type' field to 'Throwing > Link') and
  - An End Event element *E1*
- Create a Catching Link Intermediate Event element *SetNext* (open the 'Properties' dialog and set the 'Type' field to 'Catching > Link')
- Add a Sequence Flow to the target element *next=first+second*

Configure BPSim

We will use Property Parameters to define how the sequence flow forms a loop during which a Fibonacci Number will be generated. The loop mechanism is implemented via the pair of Link Events.

Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager')

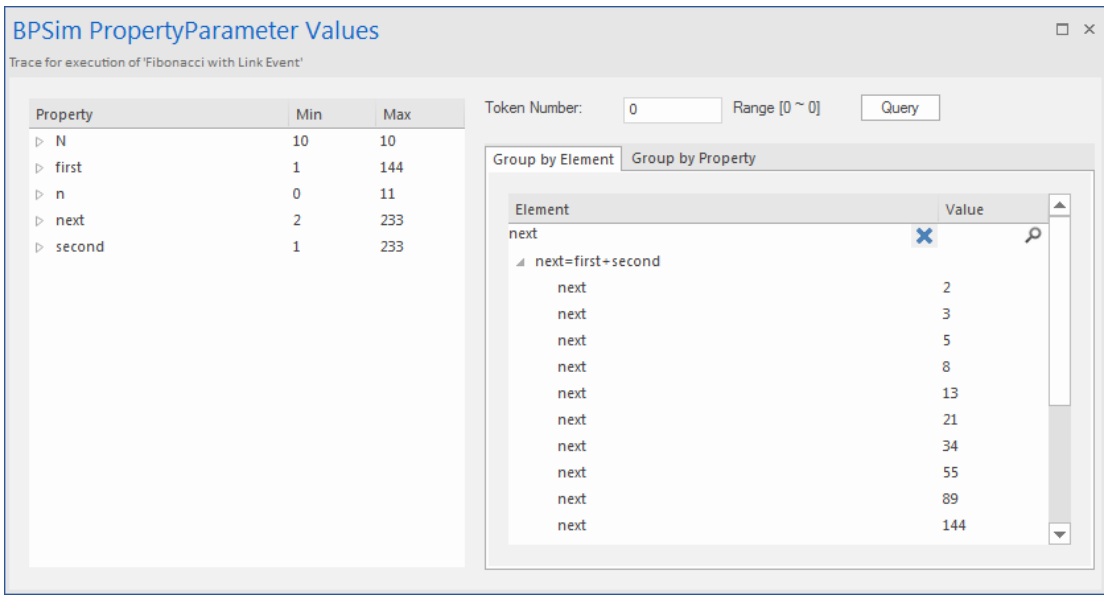
Task	Action
Element: S1	In the element type list on the left, expand the Start Event group and click on *S1*.

	Click on the 'Control Tab' and on the 'New Parameter' drop-down arrow; select 'TriggerCount'. In the 'Value' field type '1'. **Click on the 'Properties' tab** Overtype the *New Property* text to create these properties: • N - and type '10' in the 'Value' field as the total number of Fibonacci Numbers to be generated • first - and type '1' in the 'Value' field • second - and type '1' in the 'Value' field • n - and type '0' in the 'Value' field as the *n*th new Fibonacci Number
Element: next=first+second	In the element type list, expand the Activity group and click on *next=first+second*. Click on the 'Properties' tab and overtype the *New Property* text with 'next '. In the 'Value' field, click on the button, click on the 'Expression' tab and type the expression '{first}+{second}'. Click on the OK button.
Element: first=second	In the element type list, in the Activity group click on *first=second*. Click on the 'Properties' tab and overtype the *New Property* text with 'first '. In the 'Value' field, click on the button, click on the 'Expression' tab and type the expression '{second}'. Click on the OK button.
Element: second=next	In the element type list, in the Activity group click on *second=next*. Click on the 'Properties' tab and overtype the *New Property* text with 'second'. In the 'Value' field, click on the button, click on the 'Expression' tab and type the expression '{next}'. Click on the OK button.
Element: n++	In the element type list, in the Activity group click on *n++*. Click on the 'Properties' tab and overtype the *New Property* text with 'n'. In the 'Value' field, click on the button, click on the 'Expression' tab and type the expression '{n}+1'. Click on the OK button.
Conditions of Gateway	In the element type list, expand the Gateway group and the LoopNode element and click on *SetNext*. Click on the 'Control' tab and on the 'New Parameter' drop-down arrow, and select 'Condition'. In the 'Value' field, click on the button, click on the 'Expression' tab and type the expression '{n} <={N}'. Click on the OK button. Now click on *E1*. Click on the 'Control' tab and on the 'New Parameter' drop-down arrow, and select

	'Condition'. In the 'Value' field, click on the ... button, click on the 'Expression' tab and type the expression '{n} > {N}'. Click on the OK button.

Run Simulation

- On the 'Configure BPSim' dialog, in the toolbar, click on the 'Run' icon; the 'BPSim Simulation Controller' dialog displays
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'
- When the simulation is complete, click on the icon in the toolbar; the 'BPSim PropertyParameter Values' dialog displays
- Click on the Query button and on the 'Group by Element' tab, and expand 'next=first+second'; all the attributes's snapshot values are listed
- Apply a filter 'next' (right-click on the list header, select 'Toggle Filter Bar' and type 'next' under the 'Element' heading); the results will resemble this image:



Ten more Fibonacci numbers are generated:

2,3,5,8,13,21,34,55,89,144

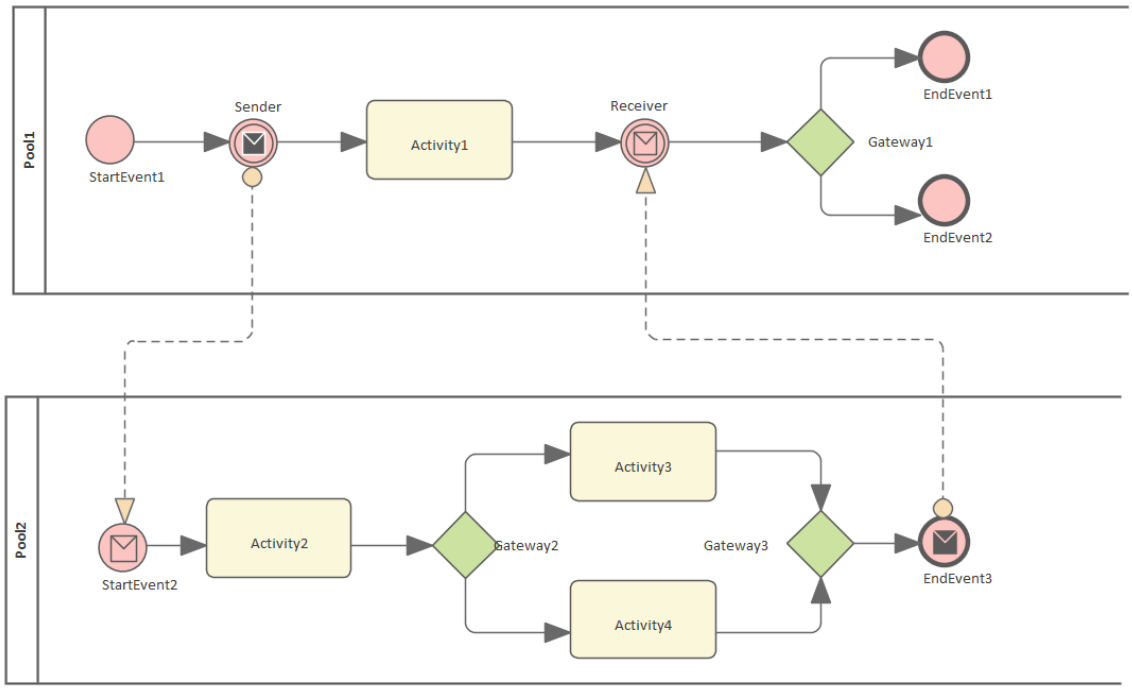
Message Event

When used in normal sequence flow, the Message Event can be used to either send or receive a message.

- When sending a message to a participant, the values of all property parameters are copied; once the message is sent the token continues along the sequence flow
- When receiving a message, the event is triggered when a message is received.

This example demonstrates the Message Event features. We will first create the BPMN model, then configure BPSim and run the simulation.

Create BPMN Model



Sequence

Pool1

- The token starts from StartEvent1
- On receiving the token, Sender (a Throwing Intermediate Message Event) creates a message and copies the current property values to the message
- Sender sends the *message* to the 'To' participant (Pool2, StartEvent2)
- Sender forwards the *token* along its sequence flow, as far as Receiver
- The token waits at Receiver until a message arrives

Pool2

- StartEvent2 receives a message and starts a token
- StartEvent2 copies the values from the message and sets these in the token
- StartEvent2 forwards the token along its sequence flow until EndEvent3
- EndEvent3 creates a message and copies the current property values to the message

- EndEvent3 sends the message to the 'To' participant (Pool1, Receiver)

Pool1 continued

- The waiting Receiver is awakened and the property values are updated from the arriving message

Create Diagram

- Create a BPMN 2.0 Collaboration diagram
- Select the option 'Create this diagram within a new CollaborationModel'
- Create *Pool1* and *Pool2* by dragging the 'Pool' icon from the Toolbox onto the diagram

Within Pool1

- Create a Start Event of type 'None', named *StartEvent1*
- Add a Sequence Flow to the target Intermediate Event of type 'Throwing Message', called *Sender*
- Add a Sequence Flow to the target Activity of type 'abstract', called *Activity1*
- Add Sequence Flows to the target:
  - End Event of type 'None', called *EndEvent1*
  - End Event of type 'None', called *EndEvent2*

Within Pool2

- Create a Start Event of type 'Message', called *StartEvent2*
- Add a Sequence Flow to the target Activity of type 'abstract', called *Activity2*
- Add a Sequence Flow to the target Gateway of type 'Exclusive', called *Gateway2*
- Add Sequence Flows to the target:
  - Activity of type 'abstract', called *Activity3*
  - Activity of type 'abstract', called *Activity4*
- Add Sequence Flows from *Activity3* and *Activity4* to the target Gateway of type 'Exclusive', called *Gateway3*
- Add a Sequence Flow to the End Event of type 'Message', called *EndEvent3*

Message Flows

- Add a Message Flow from *Sender* to *StartEvent2*
- Add a Message Flow from *EndEvent3* to *Receiver*

Configure BPSim

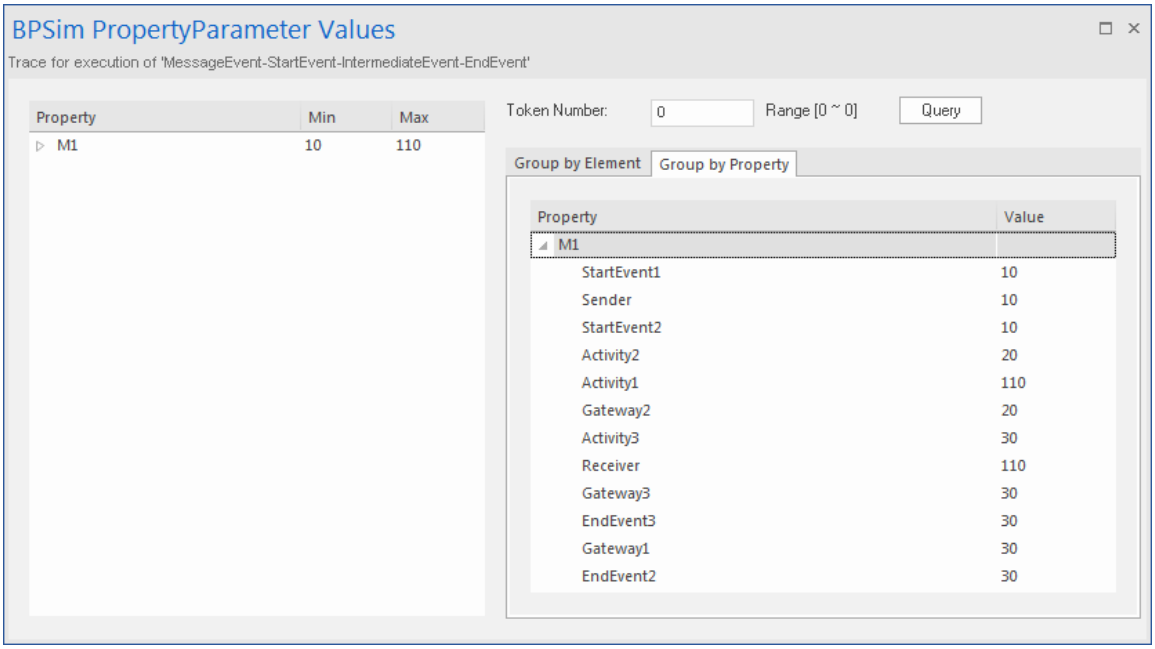
In order to show the Message Flow's ability to carry values, we create a Property Parameter 'M1' and change its value in each Activity. We then use the value of 'M1' as part of the expression for the condition of the Sequence Flow.

Task	Description
Create Artifact and Package	• Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Click on the button in the 'Select/Create Artifact' field and create an Artifact called 'MessageEvent-StartEvent-IntermediateEvent-EndEvent' • In the 'Select Package' field select the Package containing the model All the BPMN elements from the model are loaded into the Configure BPSim

	window.
Property Values	We will give the Property Parameter 'M1' an initial value of 10 at *StartEvent1*. Then we change the value as the token flows through the processes and the value is copied across participants. In the element list on the left of the dialog: • Expand the 'StartEvent' group, click on *StartEvent1* and on the 'Properties' tab, and overtype *New Property* with 'M1'; in the 'Value' field click on the ... button and select 'Constant' and 'Numeric, and type '10' in the 'Constant Numeric' field • Expand the 'Activity' group, click on *Activity1* and on the 'Properties' tab, and overtype *New Property* with 'M1'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} + 100' in the 'Expression' field • Click on *Activity2* and on the 'Properties' tab, and overtype *New Property* with 'M1'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} + 10' in the 'Expression' field • Click on *Activity3* and on the 'Properties' tab, and overtype *New Property* with 'M1'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} + 10' in the 'Expression' field • Click on *Activity4* and on the 'Properties' tab, and overtype *New Property* with 'M1'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} + 1' in the 'Expression' field *Tip: The format of* {PropertyName} *is a convenient short form of* getProperty("PropertyName").
Control Parameters	We only need one token in this simulation for evaluating the behavior of the model. • In the expanded 'StartEvent' group, click on *StartEvent1* and the 'Control' tab; click on the 'New parameter' drop-down arrow and select 'Trigger Count', and type in a 'Value' of '1' Now set up the conditions for the Gateways' outgoing Sequence Flows. In the element list on the left of the dialog, expand the 'Gateway' group: • Expand *Gateway1*, click on *EndEvent1* and on the 'Control' tab, then click on the 'New Parameter' drop-down arrow and select 'Condition'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} >= 50' in the 'Expression' field • Click on *EndEvent2* and on the 'Control' tab, then click on the 'New Parameter' drop-down arrow and select 'Condition'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} < 50' in the 'Expression' field • Expand *Gateway2*, click on *Activity3* and on the 'Control' tab, then click on the 'New Parameter' drop-down arrow and select 'Condition'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} >= 15' in the 'Expression' field • Click on *Activity4* and on the 'Control' tab, then click on the 'New Parameter' drop-down arrow and select 'Condition'; in the 'Value' field click on the ... button and select 'Expression', and type '{M1} < 15' in the 'Expression' field

Run Simulation

- On the 'Configure BPSim' dialog toolbar, click on the Run button; the 'BPSim Simulation Controller' dialog displays
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'; the simulation starts
- When the simulation is finished, click on the button; the 'Property Parameter Values' dialog displays, tracing the values for properties during simulation
- In the 'Token Number' field type '0', then click on the Query button and on the 'Group by Property' tab



Analysis

As the 'ProcessingTime' of *Activity1* was set as a distribution value, it turns out that:

- [Process1]'s 'M1' value after *Pool1.StartEvent1* is '10', as expected
- *[Process2] *Pool2.StartEvent2*'s 'M1' value is '10'; this value is carried from a message sent by *Pool1.Sender*

Now there are actually two 'M1's - *Process1.M1* and *Process2.M1*

- [Process2] *Pool2.Activity2* increased *Process2.M1* by 10; *[Process2.M1 == 20]*
- [Process1] *Pool1.Activity1* increased *Process1.M1* by 100; [*Process1.M1 == 110]*
- [Process2] The condition expressions are evaluated; as '20 > 15', the token will flow to *Activity3 [Process2.M1 == 20]*
- [Process2] *Pool2.Activity3* increased *Process2.M1* by 10; *[Process2.M1 == 30]*
- [Process1] *Pool1.Receiver* is reached and waiting *[Process1.M1 == 110]*
- [Process2] *Pool2.Gateway3* serves as a Merge node and continues to *EndEvent3 [Process2.M1 == 30]*
- *[Process1] *Pool1.Receiver is* woken up by a message(carrying *M1 == 30*) and *Process1.M1*'s value changes from 110 to 30
- [Process1] The condition expressions are evaluated; as '30 < 50', the token will flow to *EndEvent2 [Process1.M1 == 30]*

Notes

- The lines marked with an asterisk (*) are the effects of Message Flows
- The order within a process is defined; however, the order between two processes is not always predictable

- The Throwing Message Event forks another process; the Catching Message serves as thread synchronization

Signal Events

A Signal Event provides the facility of loosely coupling 'throwers' and 'catchers' by publish-subscribe integration. A 'thrower' will broadcast a signal rather than addressing it to a particular process; any process listening for that particular event could trigger a new instance using a Signal Start Event.

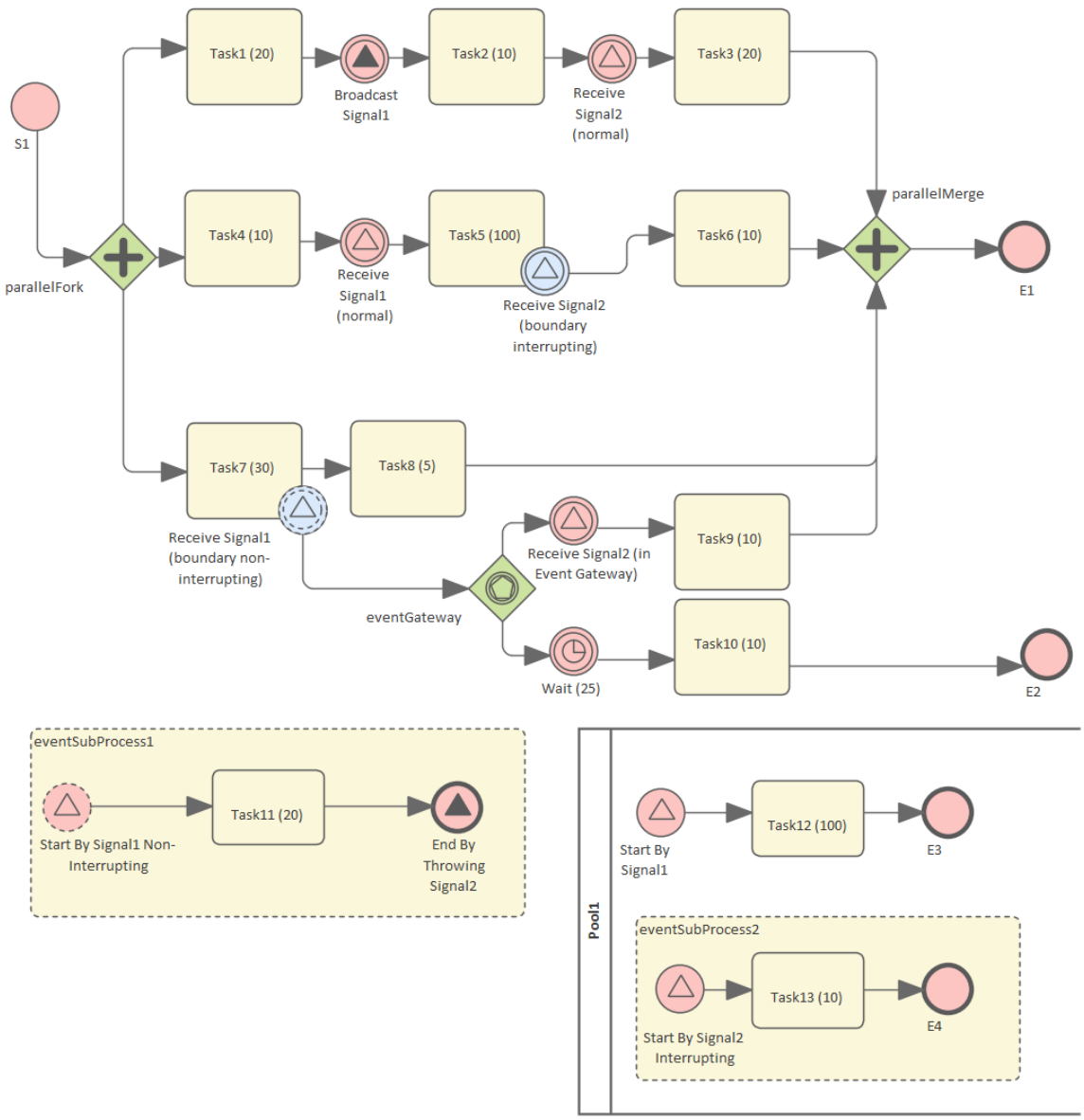
A Signal can be thrown from either a throwing Intermediate Event or a throwing End Event, and can be caught in a Start Event or a catching Intermediate Event (including a boundary Signal Event).

In this example, we demonstrate these Signal Events and their impact on the lifelines of tasks, via BPSim parameter settings.

- Start Signal Event:
  - *Start By Signal1* in top level process (Pool1)
  - *Start By Signal2* Interrupting in event sub-process *eventSubProcess2*
  - *Start By Signal1* Non Interrupting in event sub-process *eventSubprocess1*
- Throwing Intermediate Signal Event:
  - *Broadcast Signal1*
- Catching Intermediate Signal Event:
  - *Receive Signal1 (normal)*
  - *Receive Signal2 (normal)*
  - *Receive Signal2 (boundary Interrupting)*
  - *Receive Signal1 (boundary non-interrupting)*
  - *Receive Signal2 (in Event Gateway)*
- End Signal Event:
  - *End By Throwing Signal2*

Create BPMN Model

In order to demonstrate the ability to communicate across processes via a Signal Event, we create a Collaboration model with a main Pool and a process in another Pool (*Pool1*).



Create the Collaboration and main process

Create a new BPMN2.0 Collaboration diagram called *CollaborationForTestingSignalEvents*, (choose the option 'Create this diagram within a new Collaboration Model'). Right-click on the diagram name in the Browser window and select the 'Encapsulate Process' option.

A Pool *PoolMain* and a process *BusinessProcess_PoolMain* are created, and these tags are set with the automatic values:

- *CollaborationForTestingSignalEvents.mainPool* is set to *PoolMain*
- *PoolMain.processRef* is set to *BusinessProcess_PoolMain*

Create the elements for the main process

Create a Start Event *S1* and add a Sequence Flow to a Fork Parallel Gateway *parallelFork*

Add Sequence Flows to:

- An Abstract Task *Task1 (20)* and then add this chain of Sequence Flows:
  - To a Throwing Intermediate Signal Event *Broadcast Signal1*
  - Then to an Abstract Task *Task2 (10)*
  - Then to a Catching Intermediate Signal Event *Receive Signal2 (normal)*
  - Then to an Abstract Task *Task3 (20)*

  - Then to a Merge Parallel Gateway *parallelMerge*
  - Then to an End Event *E1*

- An Abstract Task *Task4 (10)* and then add this chain of Sequence Flows:
  - To a Catching Intermediate Signal Event *Receive Signal1 (normal)*
  - Then to an Abstract Task *Task5 (100),* on which you create a Boundary Interrupting Catching Intermediate Signal Event *Receive Signal2 (boundary interrupting)*
  - Then to an Abstract Task *Task6 (10)*
  - Then to the earlier Merge Parallel Gateway *parallelMerge*
- An Abstract Task *Task7 (30)*, and then add this chain of Sequence Flows:
  - To an Abstract Task *Task8 (5)*
  - Then to the earlier Merge Parallel Gateway *parallelMerge*

On *Task7 (30)*, create a Boundary Non-interrupting Catching Intermediate Signal Event *Receive Signal1 (boundary non-interrupting)*. Add a Sequence Flow to an Event Gateway *eventGateway*, and to that add Sequence Flows to:

- A Catching Intermediate Signal Event *Receive Signal2 (in Event Gateway),* and then this chain of Sequence Flows:
  - To an Abstract Task *Task9 (10)*
  - Then to the earlier Merge Parallel Gateway *parallelMerge*
- A Catching Intermediate Timer Event *Wait (25)*, and then this chain of Sequence Flows:
  - To an Abstract Task *Task10 (10)*
  - Then to an End Event *E2*

Create an Event sub-process (triggered by a non-interrupting Start Signal Event) in the main process

- Create an Activity *eventSubProcess1* and, in its 'Properties' dialog, set the 'Type' field to *subProcess* and change the attribute 'triggeredByEvent' to *true*
- Within *eventSubProcess1* create a Start Event *Start By Signal1 Non Interrupting* and, in its 'Properties' dialog, set the 'Type' field to *Event Sub-Process Non-Interrupting > Signal*
- Add a Sequence Flow to a target Abstract Task *Task11 (20)*
- Add a Sequence Flow to a target End Event *End By Throwing Signal2* and, in the element 'Properties' dialog, set the 'Type' field to *Signal*

Create another process

- From the Toolbox, drag and drop the 'Pool' icon onto the diagram and name the element *Pool1*
- Right-click on *Pool1* in the Browser window and select the 'Encapsulate Process' option; a process *BusinessProcess_Pool1* is created and the tag 'Pool1.processRef' is set to *BusinessProcess_Pool1*

Create the main process for *Pool1*

- Create a Signal Start Event *Start By Signal1*
- Add a Sequence Flow to a target Abstract Task *Task12 (100)*
- Add a Sequence Flow to a target End Event *E3*

Create an Event sub-process to interrupt *Pool1*

- Create an Activity *eventSubProcess1* and, in the 'Properties' dialog, set the 'Type' field to *subProcess*; change the attribute 'triggeredByEvent' to *true*
- Within *eventSubProcess2* create a Start Event *Start By Signal2 Interrupting* and, in the 'Properties' dialog, set the 'Type' field to *Event Sub-Process Interrupting > Signal*
- Add a Sequence Flow to a target Abstract Task *Task13 (10)*
- Add a Sequence Flow to a target End Event *E4*

Create the BPMN2.0 Signal Elements and configure for Signal Events

In the BPMN 2.0 Toolbox, expand the 'BPMN 2.0 - Types' page and drag the 'Signal' icon onto the diagram; name the element *Signal1*. Drag the icon onto the diagram again to create *Signal2*. These are root elements (which can be used by all processes) so they will be created directly under the model Package.

Double-click on each of the Signal Event elements and, in the 'Value' field for the 'signalRef' tag, click on the button and browse to the appropriate Signal element.

Tips: Alternatively, you can drag the Signal element from the Browser window and drop it on the Event elements in the diagram; a context menu displays, from which you select the 'set signalRef' option.

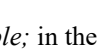
- Set signalRef to 'Signal1' on:
  - *Broadcast Signal1*
  - *Start By Signal1* in top level process (*Pool1*)
  - *Start By Signal1 Non Interrupting* in Event sub-process *eventSubprocess1*
  - *Receive Signal1 (normal)*
  - *Receive Signal1 (boundary non-interrupting)*
- Set signalRef to 'Signal2' on:
  - *Start By Signal2 Interrupting* in Event sub-process *eventSubProcess2*
  - *Receive Signal2 (normal)*
  - *Receive Signal2 (boundary Interrupting)*
  - *Receive Signal2 (in Event Gateway)*

Configure BPSim

In this section, we create the Configuration Artifact, specify the model Package and configure the parameter values of each element.

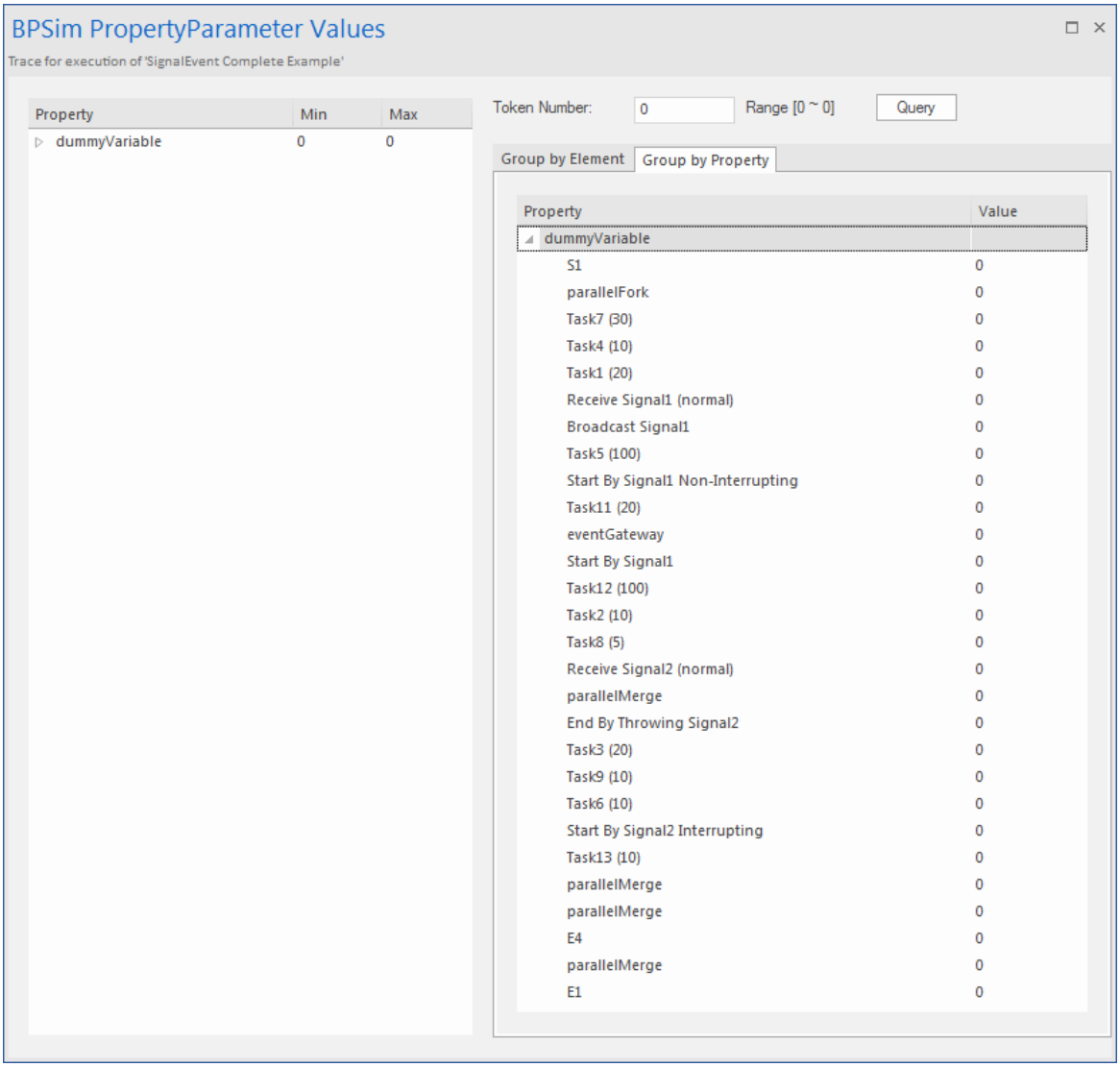
The configuration is quite simple because none of the Signal Events require any BPSim configurations. All we have to do is set the processing time for tasks so we can observe how processes, threads and tasks are started and interrupted.

Task	Description
Set up configuration	• Open the Configure BPSim window ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'SignalEvent Complete Example' (in the 'Select/Create Artifact' field, click on the button and select its parent Package and click on the Add New button, then type in the element name and click on the Save button and the OK button) Then all the BPMN elements will be loaded in to the Configure BPSim window.
Non-Signal Events	• In the element list on the left of the dialog, expand the 'StartEvent' group, then click on *S1* and on the 'Control' tab; click on the 'New Parameter' drop-down arrow and select 'Trigger Count', then type '1' in the 'Value' field • Expand the 'IntermediateEvent' group, then click on *Wait (25)* and on the 'Control' tab; click on the 'New Parameter' drop-down arrow and select 'InterTriggerTimer', then click on the button in the 'Value' field; select 'Constant' and 'Numeric', and type '25' in the 'Constant Numeric' field and 'seconds' in the 'TimeUnit' field
Dummy variable for Process	The simulation controller displays a list showing the runtime token count for each element. For example, 4 tokens have passed the Gateway element *parallelMerge* in a simulation. This is quite useful for certain statistics and analysis. However, it does not show WHEN *parallelMerge* was traversed during the simulation. In order to get the exact trace for a single token we use the property trace utility, which relies on property parameters. So we create a dummy parameter.

	In the 'Configuration BPSim' dialog, expand the 'BusinessProcess' group. • Click on *BusinessProcess_Main* and on the 'Properties' tab, and overtype *New Property* with *dummyVariable;* in the 'Value' field, click on the  button and on 'Constant' and 'Numeric', and in the 'Constant Numeric' field type '0' • Click on *BusinessProcess_Pool1* and perform exactly the same actions as for *BusinessProcess_Main*
Processing Time for Tasks	Expand the 'Activity' group and for each Task element listed here: select the 'Time' tab, click on the 'New Parameter' drop-down arrow and select 'ProcessingTime', then click on the button on the 'Value' column, select 'Constant' and 'Numeric', type the value as indicated into the 'Constant Numeric' field and select 'seconds' in the 'TimeUnit' field. • Task1 (20): 20 seconds • Task2 (10): 10 seconds • Task3 (20): 20 seconds • Task4 (10): 10 seconds • Task5 (100): 100 seconds • Task6 (10): 10 seconds • Task7 (30): 30 seconds • Task8 (5): 5 seconds • Task9 (10): 10 seconds • Task10 (10): 10 seconds • Task11 (20): 20 seconds • Task12 (100): 100 seconds • Task13 (10): 10 seconds

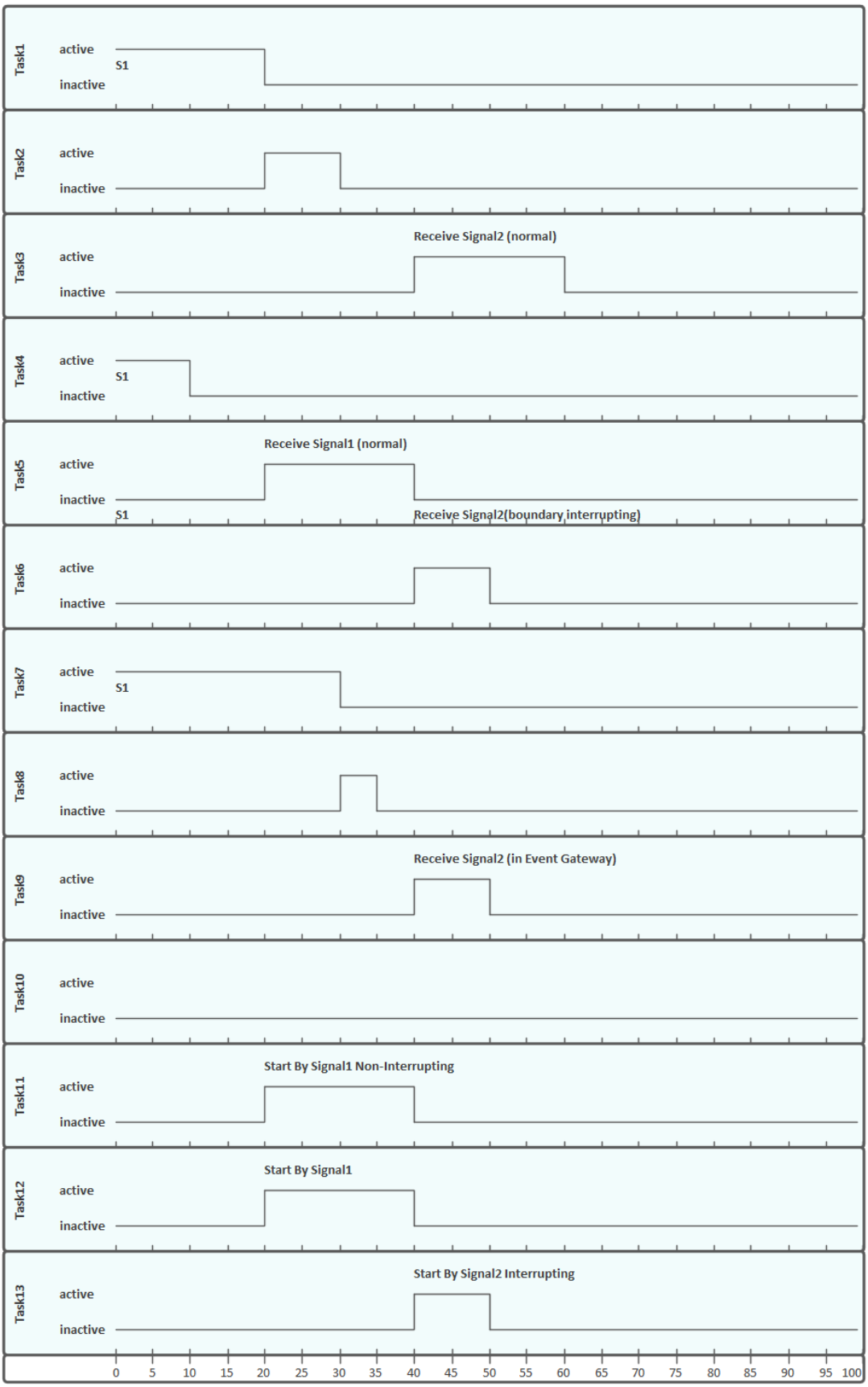
Run Simulation

- On the 'Configure BPSim' dialog Toolbar, click on the 'Run' icon to open the 'BPSim Simulation Controller' dialog
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'
- After simulation, click on the button on the tool bar to display the 'BPSim PropertyParameter Values' dialog
- Click on the Query button and on the 'Group by Property' tab, and expand 'dummyVariable'



Analysis

From the direct results of the simulation it might not be obvious what has taken place; however, if we draw the lifeline for each task, it becomes quite clear.



- *Task1*, *Task4* and *Task7* started in parallel

- *Task2* started immediately after *Task1* finished (without stopping at the throwing event)
- At 20 seconds, *Signal1* was broadcast by the Throwing Intermediate Event *Broadcast Signal1* and:
  - *Receive Signal1 (normal)* was activated and *Task5* started
  - *Start By Signal1 Non-Interrupting* was activated and *Task11* in *eventSubProcess1* started
  - *Start By Signal1* was activated and *Task12* in *Pool1 s*tarted
- At 40 seconds, *Signal2* was broadcast by the End Event *End By Throwing Signal2* and:
  - *Receive Signal2 (normal)* was activated and *Task3* started
  - *Task5* was interrupted and *Task6* started
  - *Receive Signal2 (in Event Gateway)* was activated and *Task9* started
  - *Start By Signal2 Interrupting* was activated, and:
    > The main process in *Pool1* was interrupted and *Task12* stopped
    > *Task13* in *eventSubProcess2* started
- The *eventSubProcess2* inside *BusinessProcess_Pool1* finished when *E4* was reached at 50 seconds
- The *BusinessProcess_MainPool* finished when *E1* was reached at 60 seconds
- The Intermediate Timer Event *Wait (25)* did not get activated because the signal event in the Gateway was activated first; as a result, *Task10* was never started

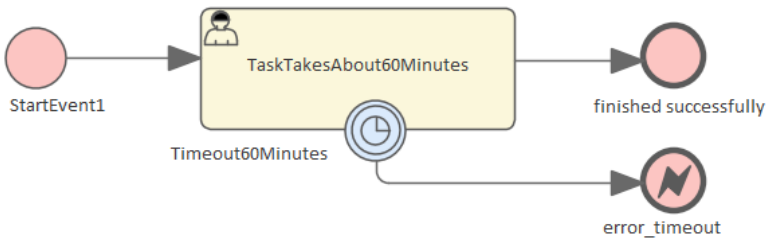
Note: The actual running time for each task can be observed from the generated BPSimReport element, by:

1. Double-clicking on the <<BPSimReport>> element.
2. Expanding the 'Time' group.
3. Expanding the task element.
4. Checking 'Total Time In Task'.

For example, for element *Task5 (100)*, although we set its processingTime to be 100 seconds, the **Total Time In Task** was 20 seconds, which was interrupted by *Receive Signal2 (boundary interrupting)* at 20 seconds.

Timer Event - Boundary

Create BPMN Model



- Create a Start Event *StartEvent1*
- Add a Sequence Flow to a target userTask *TaskTakesAbout60Minutes*
- Add a Sequence Flow to a target End Event *finished successfully*
- Create an Intermediate Event, dragging the icon from the Toolbox and dropping it onto *TaskTakesAbout60Minutes;* select 'Edge-Mounted' and 'Timer' from the automatic menus, and call the element *Timeout60Minutes*
- Add a Sequence Flow to a target End Event (Error) *error_timeout*

Configure BPSim

In this section, we create the Configuration Artifact, identify the parent Package and set the parameter values of each element.

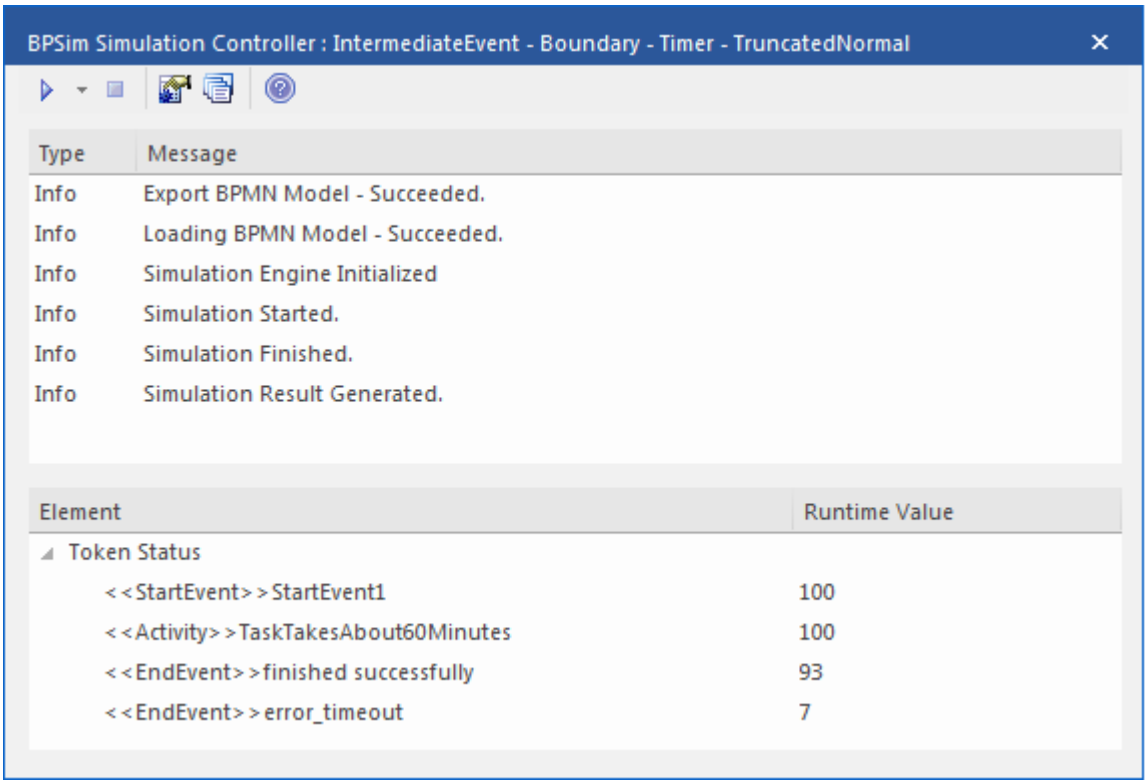
Objects	Action
Create Artifact and Package	• Open the 'Configure BPSim' dialog ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'IntermediateEvent - Boundary - Timer - TruncatedNormal' (in the 'Select/Create Artifact' field, click on the ... button and select its parent Package, click on the Add New button, then type in the element name and click on the Save button and the OK button) Then all the BPMN elements will be loaded in to the 'Configure BPSim' dialog.
StartEvent1	In the element list on the left of the dialog, expand the 'StartEvent' group and click on *StartEvent1*. • Click on the 'Control' tab • Click on the 'New Parameter' drop-down arrow and select 'TriggerCount' • In the 'Value' field type '100'
TaskTakesAbout60Minutes	In the element list on the left of the dialog, expand the 'Activity' group and click on *TaskTakesAbout60Minutes*. • Click on the 'Time' tab • Click on the 'New Parameter' drop-down arrow and select 'ProcessingTime' • In the 'Value' field click on the ... button and select 'Distribution' and 'TruncatedNormal'

	• In the 'Mean' field type '50' • In the 'StandardDeviation' field type '10' • In the 'Min' field type '0' • In the 'Max' field type '1000' • Click on the OK button
Timeout60Minutes	In the element list on the left of the dialog, expand the 'IntermediateEvent' group and click on *Timeout60Minutes*. • Click on the 'Control' tab • Click on the 'New Parameter' drop-down arrow and select 'InterTriggerTimer' • Set the value to '000:000:000 001:00:00' (that is, 1 hour)

Run Simulation

- On the Configure BPSim window toolbar, click on the 'Run' icon to open the 'BPSim Simulation Controller' dialog
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'
- After simulation, click on the button on the tool bar to display the 'BPSim PropertyParameter Values' dialog
- Click on the Query button and on the 'Group by Property' tab, and expand 'dummyVariable'

On simulation, we get this result:



Analysis

Since the ProcessingTime of *TaskTakesAbout60Minutes* was set as a distribution value, it turns out that:

- 93 out of 100 finished in 1 hour, so the normal flow to *finished successfully* takes effect

- 7 out of 100 finished in more than 1 hour, so the exception flow to *error_timeout* takes effect

Other Configurations

In the example folder, there are two other Business Process Simulation Artifacts that set the ProcessingTime as a constant value of 50 minutes and 80 minutes, other settings remain the same.

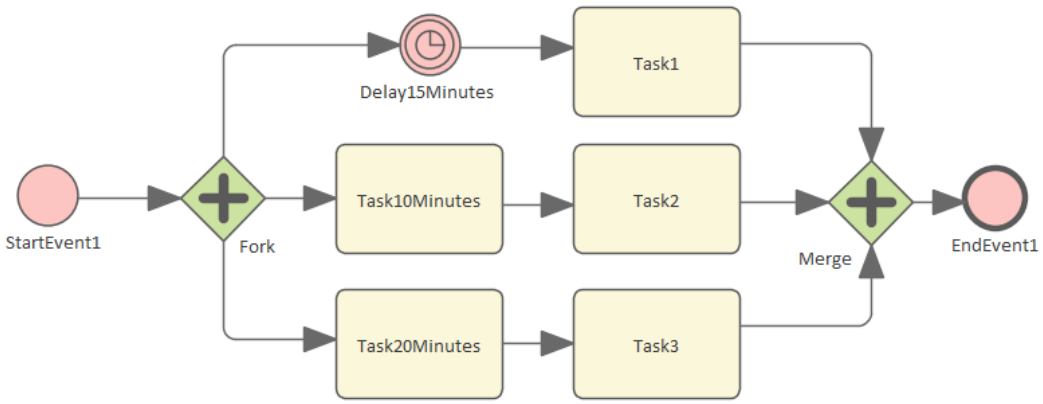
Run simulation on these two Artifacts:

- ProcessingTime configured to be 50 minutes always finishes in normal flow
- ProcessingTime configured to be 80 minutes always finishes in exception flow

Timer Event - Standalone Intermediate Event

When a Timer Intermediate Event is used within the normal sequence flow as a standalone element, it acts as a delay mechanism.

Create BPMN Model



- Create a Start Event called *StartEvent1*
- Add a Sequence Flow to a target Parallel Gateway called *Fork*
- Add Sequence Flows to:
  - A Standalone Timer Intermediate Event called *Delay15Minutes*, and from that a Sequence Flow to an Activity called *Task1*
  - An Activity called *Task10Minutes*, and from that a Sequence Flow to an Activity called *Task2*
  - An Activity called *Task20Minutes*, and from that a Sequence Flow to an Activity called *Task3*
- From *Task1*, *Task2* and *Task3* create Sequence Flows to a Merge Parallel Gateway called *Merge*
- Add a Sequence Flow to a target EndEvent called *EndEvent1*

Configure BPSim

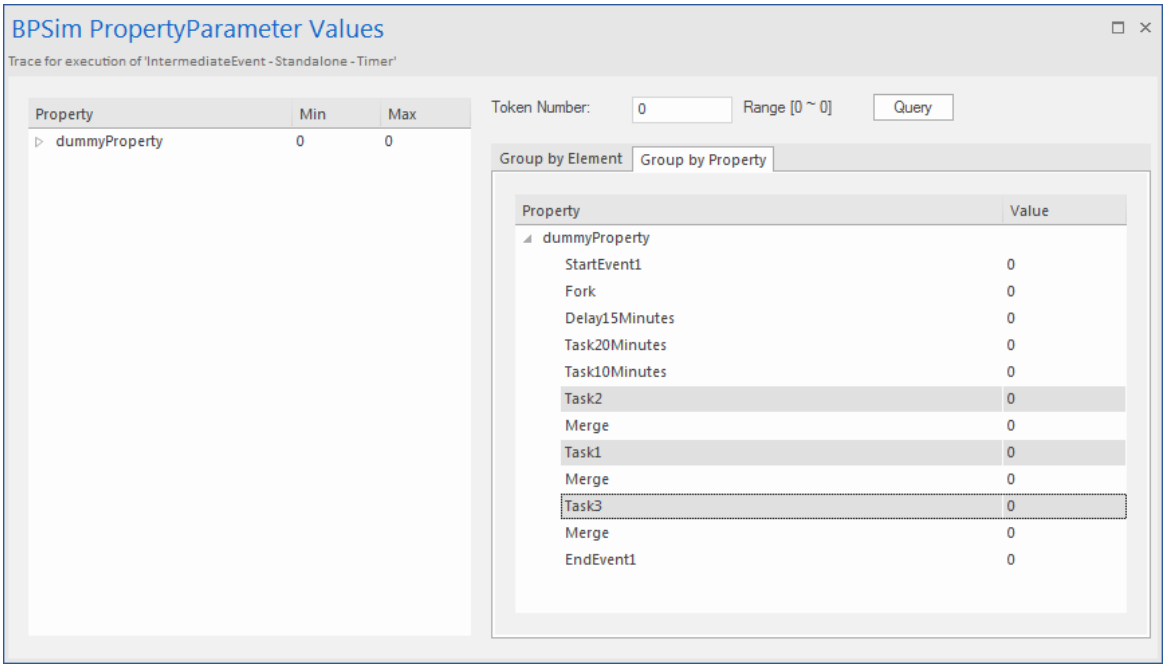
In this section, we create the Configuration Artifact, specify the model Package and configure the parameter values of each element.

Object	Action
Create Artifact and Package	• Open the 'Configure BPSim' dialog ('Simulate > Process Analysis > BPSim > Open BPSim Manager') • Create an Artifact named 'IntermediateEvent - Standalone - Timer' (in the 'Select/Create Artifact' field, click on the ... button and select its parent Package and click on the Add New button, then type in the element name and click on the Save button and the OK button) Then all the BPMN elements will be loaded in to the 'Configure BPSim' dialog.
StartEvent1	• In the element list on the left of the dialog, expand the 'StartEvent' group, then click on *StartEvent1* and on the 'Control' tab • Click on the 'New Parameter' drop-down arrow and select 'Trigger Count', then

	type '1' in the 'Value' field • Click on the 'Properties' tab • Overtype *New Property* with *dummyProperty;* in the 'Value' field, click on the ... button and on 'Constant' and 'Numeric', and in the 'Constant Numeric' field type '0' With this property, the 'Property Trace' dialog will be able to show the sequence of element flows during simulation.
Delay15Minutes	• In the element list on the left of the dialog, expand the 'IntermediateEvent' group, then click on *Delay15Minutes* and on the 'Control' tab • Click on the 'New Parameter' drop-down arrow and select 'InterTriggerTimer', then set the 'Value' field to 15 minutes ('000:000:000 000:15:00')
Task10Minutes	• In the element list on the left of the dialog, expand the 'Activity' group, then click on *Task10Minutes* and on the 'Time' tab • Click on the 'New Parameter' drop-down arrow and select 'ProcessingTime', then set the 'Value' field to 10 minutes ('000:000:000 000:10:00')
Task20Minutes	• In the element list on the left of the dialog, expand the 'Activity' group, then click on *Task20Minutes* and on the 'Time' tab • Click on the 'New Parameter' drop-down arrow and select 'ProcessingTime', then set the 'Value' field to 20 minutes ('000:000:000 000:20:00')

Run Simulation

- On the 'Configure BPSim' dialog Toolbar, click on the 'Run' icon to open the 'BPSim Simulation Controller' dialog
- Click on the 'Run' icon drop-down arrow and select 'Standard Simulation'
- After simulation, click on the button on the tool bar to display the 'BPSim PropertyParameter Values' dialog
- Click on the Query button and on the 'Group by Property' tab



Analysis

The *Fork* Parallel Gateway will activate the outgoing Sequence Flows simultaneously (the order is undefined and not important). However, we would expect the order of the Tasks to be exactly:

- *Task2*
- *Task1*
- *Task3*

This order is determined by the BPSim parameters set on two of the Activities (ProcessingTime) and the Timer Intermediate Event (InterTriggerTimer). The sequence shown in the 'BPSim PropertyParameter Values' dialog confirms that *Task2* comes ahead of *Task1*, which comes ahead of *Task3*.

Paint Wall Process Simulation (Call Activity)

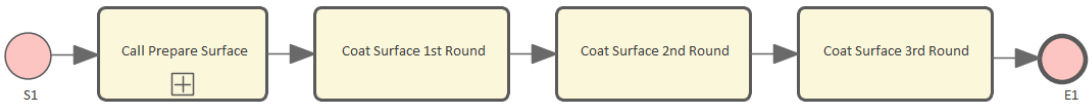
This is a simple example to simulate the process of painting a wall. We define the main process as preparing the surface and then painting it three times. Preparing the surface is further divided into tasks such as sanding and cleaning.

We assume that applying each of the three coats of paint is the same process, except that the time randomly spent on each coat might be different.

Create BPMN Model

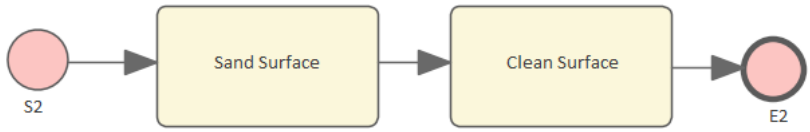
This simulation operates on two processes.

The main process - Paint Wall Process



1. Create a Start Event called *S1*.
2. Add a Sequence Flow to a target callProcessActivity called *Call Prepare Surface.*
3. Add a Sequence Flow to a target callGlobalTaskActivity called *Coat Surface 1st Round.*
4. Add a Sequence Flow to a target callGlobalTaskActivity called *Coat Surface 2nd Round.*
5. Add a Sequence Flow to a target callGlobalTaskActivity called *Coat Surface 3rd Round.*
6. Add a Sequence Flow to a target End Event called *E1*.

The re-used process - Prepare Surface Process



1. Create a Start Event called *S2*.
2. Add a Sequence Flow to a target Abstract Task called *Sand Surface.*
3. Add a Sequence Flow to a target Abstract Task called *Clean Surface.*
4. Add a Sequence Flow to a target End Event called *E2*.

Set Global Task and re-used process to call Activities

1. Create a Global Task Activity called *Coat Surface.*
2. Double-click on each of *Coat Surface 1st Round, Coat Surface 2nd Round* and *Coat Surface 3rd Round*, and set the tag 'calledActivityRef' to *Coat Surface.*
   Tip: You can also drag the Global Task 'Coat Surface' from the Browser window and drop it on the Call Activity element, clicking the 'Set calledActivityRef' option on the context menu.
3. Double-click on *Call Prepare Surface* and set the tag 'calledActivityRef' to *Prepare Surface Process.*
   Tip: You can also drag the process 'Prepare Surface Process' from the Browser window and drop it on the Call Activity element, clicking the 'Set calledActivityRef' option on the context menu.

Configure BPSim

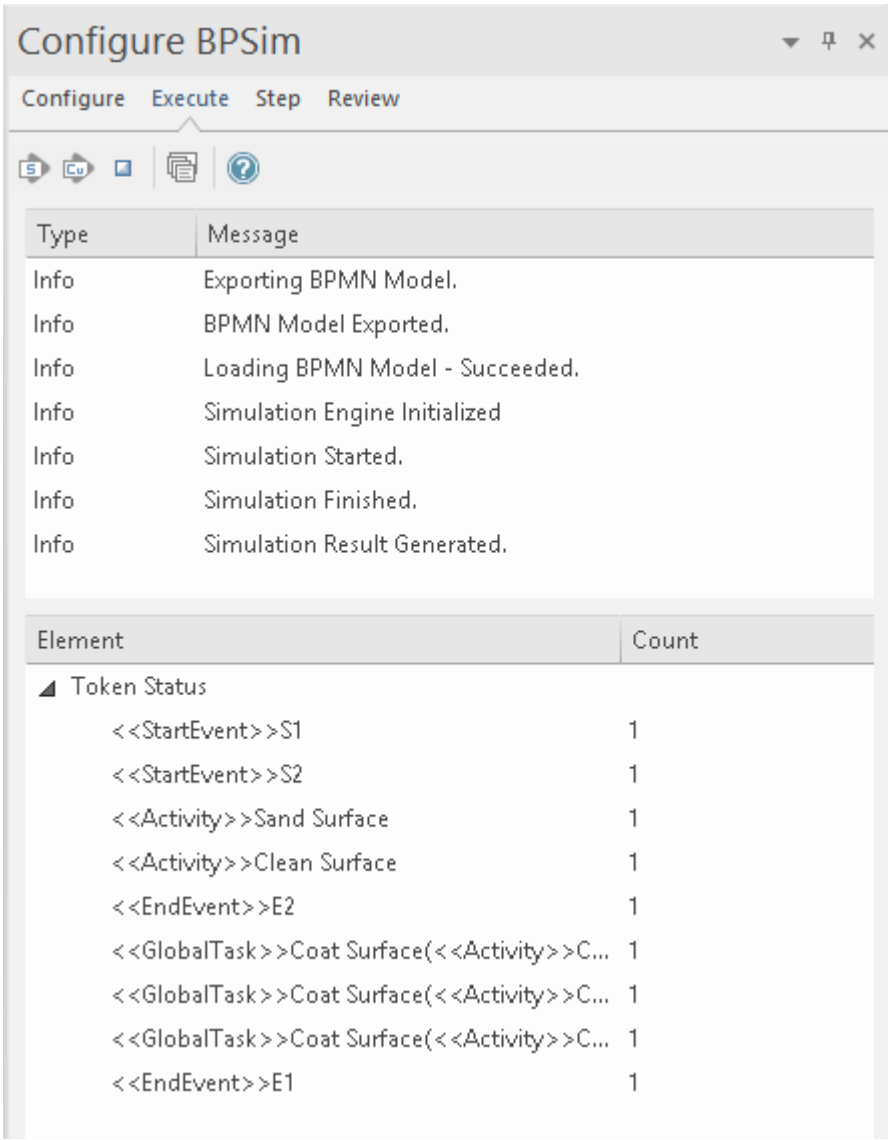
1. Open the 'Configure BPSim' dialog ('Simulate > Process Analysis > BPSim > Open BPSim Manager').
2. Click on the icon and create a Business Process Simulation Artifact named *Paint Wall Simulation.*
3. Click on the icon and select the Package containing the corresponding BPMN 2.0 model.

Object	Activity
Fixed Scaling Time	1. Open the 'Prepare Surface Process' diagram. 2. Click on the Activity *Sand Surface* and, on the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Time parameter called 'Processing Time'. 3. In the 'Values' field, change the setting to 0 00:30:00 (that is, 30 minutes). 4. Click on the Activity *Clean Surface* and, on the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Time parameter called 'Processing Time'. 5. In the 'Values' field, change the setting to 0 00:10:00, (that is, 10 minutes). 6. Click on the icon.
Random Coating Time	1. In the Browser window, click on the Global Task Activity *Coat Surface.* 2. In the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Time parameter called 'Processing Time'. 3. In the 'Values' field click on the button, and on the parameter dialog click on the 'Distribution' tab and on 'Poisson'. 4. In the 'Mean' field type '10', then click on the OK button. 5. Click on the icon. With this setting, the mean value of the random numbers generated by the Poisson distribution is 10. If you prefer, you can choose other types of distribution.
TriggerCount on S1	On the 'Paint Wall Process' diagram click on the Start Event *S1.* 1. In the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Control parameter called 'TriggerCount'. 2. In the 'Values' field type '1'. 3. Click on the icon.

Run Simulation

1. On the 'Execute' tab of the Configure BPSim window, click on the icon.

When the simulation is complete, it provides a result similar to this:



Flow Analysis

For the only token started on *S1*, we can see from the Configure BPSim window 'Execute' tab how the flow develops:

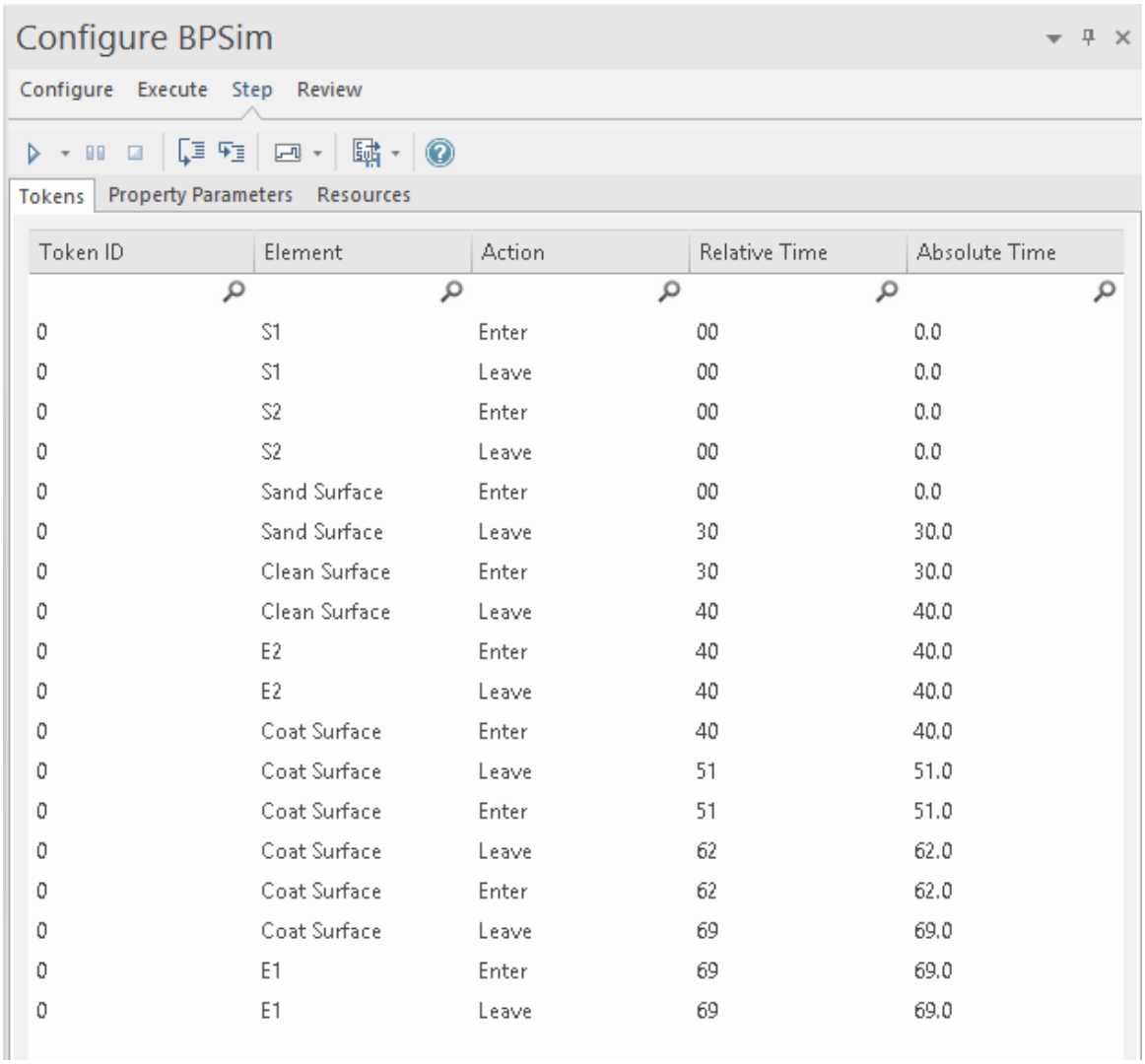
Element	Count
Token Status	
<<StartEvent>>S1	1
<<StartEvent>>S2	1
<<Activity>>Sand Surface	1
<<Activity>>Clean Surface	1
<<EndEvent>>E2	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 1st Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 2nd Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 3rd Round)	1
<<EndEvent>>E1	1

- When reaching the callProcessActivity, the called process is activated; so we have *S2 ~ E2*

- When reaching a callGlobalTaskActivity, the called Global Task is activated - the notation reads: *GlobalTask name (called activity name)*; the global *Coat Surface* was called three times:
  - *Coat Surface(Coat Surface 1st Round)*
  - *Coat Surface(Coat Surface 2nd Round)*
  - *Coat Surface(Coat Surface 3rd Round)*

Time Analysis

Click on the 'Steps' tab of the Configure BPSim window, and on the Tokens tab, which resembles this illustration:



Token ID	Element	Action	Relative Time	Absolute Time
0	S1	Enter	00	0.0
0	S1	Leave	00	0.0
0	S2	Enter	00	0.0
0	S2	Leave	00	0.0
0	Sand Surface	Enter	00	0.0
0	Sand Surface	Leave	30	30.0
0	Clean Surface	Enter	30	30.0
0	Clean Surface	Leave	40	40.0
0	E2	Enter	40	40.0
0	E2	Leave	40	40.0
0	Coat Surface	Enter	40	40.0
0	Coat Surface	Leave	51	51.0
0	Coat Surface	Enter	51	51.0
0	Coat Surface	Leave	62	62.0
0	Coat Surface	Enter	62	62.0
0	Coat Surface	Leave	69	69.0
0	E1	Enter	69	69.0
0	E1	Leave	69	69.0

You can check timing in the list as it is, but to make the process easier type 'Leave' in the Filter bar field of the 'Action' column to show only the records containing that text string in that column.

Tokens | Property Parameters | Resources

Token ID	Element	Action	Relative Time	Absolute Time
		Leave		
0	S1	Leave	00	0.0
0	S2	Leave	00	0.0
0	Sand Surface	Leave	30	30.0
0	Clean Surface	Leave	40	40.0
0	E2	Leave	40	40.0
0	Coat Surface	Leave	51	51.0
0	Coat Surface	Leave	62	62.0
0	Coat Surface	Leave	69	69.0
0	E1	Leave	69	69.0

The report displays as shown, and we can make this analysis:

- The Call Activity *Call Prepare Surface* took 40 minutes, composed of *Sand Surface* (30 minutes) and *Clean Surface* (10 minutes), as defined
- *Coat Surface (1st Round)* took 11 minutes; *Coat Surface (2nd Round)* took 11 minutes; *Coat Surface (3rd Round)* took 7 minutes - the figures 11, 11, 7 are randomly generated by the Poisson(10) distribution; what is important here is that each call instance of the Global Task has its own values
- *Coat Surface* has a total time collected from all instances: 11 + 11 + 7 = 29
- The Sum Processing Time for the *Paint Wall Process* is 69 minutes, composed of the four Call Activities: 40 + 11 + 11 + 7 = 69

Customized Simulation

We can configure a 'Result Request' on BPMN elements to customize the simulation report so that we only report on the parameters we are interested in.

Configure Result Request

1. On the 'Paint Wall Process' diagram click on the Activity *Coat Surface 1st Round*.
2. In the Configure BPSim window, click on the *New Parameter* drop-down arrow and create a Time parameter called 'ProcessingTime'.
3. Click on the toolbar icon. The 'Result Request' column displays to the right of the 'Parameter' column; click on the drop-down arrow and select the 'sum' checkbox. Click on the OK button.
4. In the 'Values' field type '1'.
5. Click on the icon.
6. Repeat steps 1 to 5 for the Activities *Call Prepare Surface*, *Coat Surface 2nd Round*, *Coat Surface 3rd Round*

- Expand the 'Business Process' group and repeat these steps for *Paint Wall Process*

Run Simulation

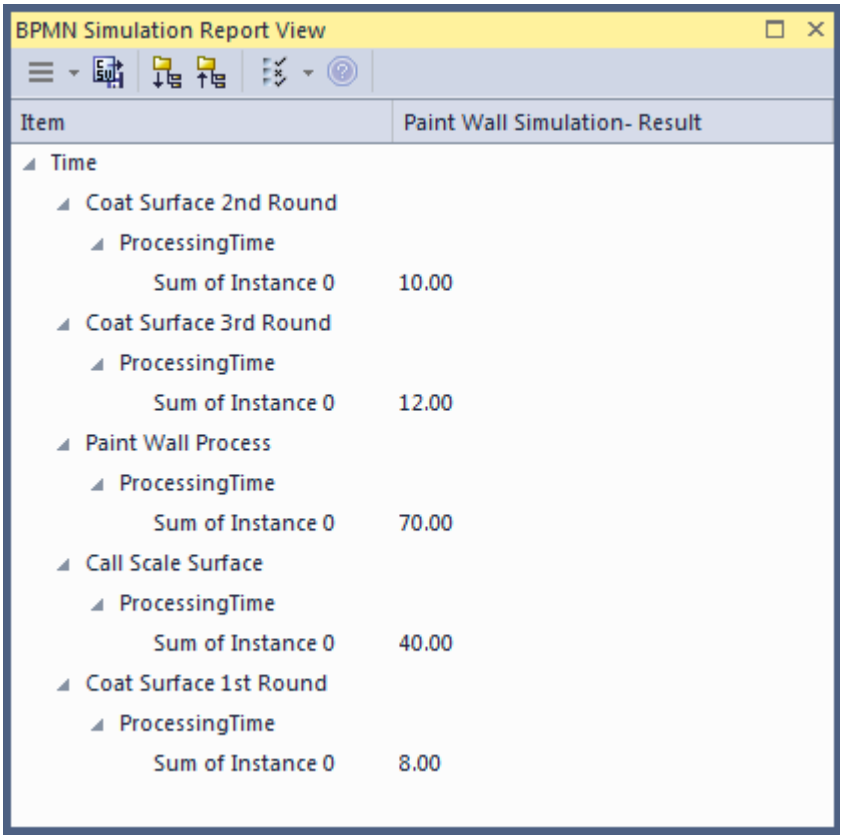
- On the 'Configure BPSim' dialog toolbar, click on the Run button; the 'BPSim Simulation Controller' dialog displays
- Click on the Run button drop-down arrow and select, in this case, 'Customized Simulation'

Flow Analysis

The Flow Analysis is exactly the same as for a Standard Simulation.

Time Analysis

On the 'BPSim Simulation Controller' dialog toolbar, click on the button; the 'BPMN Simulation Report View' displays.



The Time Analysis is the same as for a Standard Simulation; however, the report only contains the 'sum' results we requested.

Note: Currently, in the Time Analysis, we can not request ProcessingTime either on the called process itself or on the Activities contained by the called process. If you have this requirement, use the Standard Simulation.

BPSim Cost Parameters

BPSim 1.0 provides the means to set cost parameters and receive cost statistics from process simulation experiments. BPSim provides a framework to determine **variable** costs according to two parameters, both related to the level of activity performed in the simulated process. These parameters are:

- Completion Cost ('Fixed Cost' in the BPSim specification) - The cost incurred whenever an operation is completed; this cost can be related to Task, Process, Sub-Process, Call Activity or Resource elements
- Time Cost ('Unit Cost' in the BPSim specification) - The cost incurred whenever a task, process, sub-process, call activity or resource is busy for a period of time

The cost parameters are supported on Activities, resources and processes. On:

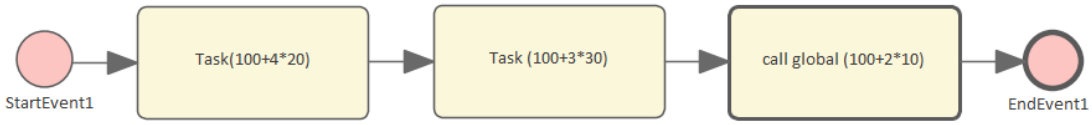
- An Activity, Completion Cost and Time Cost (Unit Cost * Time) are both incurred whenever an Activity finishes
- A Resource, Completion cost and Time Cost are both incurred whenever each involved resource finishes an Activity
- A Process, Completion cost and Time Cost are both incurred whenever a process finishes

Those costs that are known without the need for simulation - for example, overall labor employment costs - are not supported by BPSim.

The configuration and simulation of cost parameters are demonstrated by two examples:

- Set Cost Parameters on Activity
- Set Cost Parameters on Resource

Set Cost Parameters on Activity



Create the BPMN Model (Activities)

1. In the Browser window, create a *StartEvent1*, a *GlobalTask1*, two AbstractTasks, and an *EndEvent1*.
2. Ctrl+drag the elements from the Browser window onto a diagram, pasting *GlobalTask1* as an Invocation (Call Activity) called *call global (100+2*10)*.
3. Give the elements names and connect them with Sequence Flows; the two AbstractTasks should be called:
   - *Task (100+3*30)* and
   - *Task (100+4*20)*.

BPSim Configuration

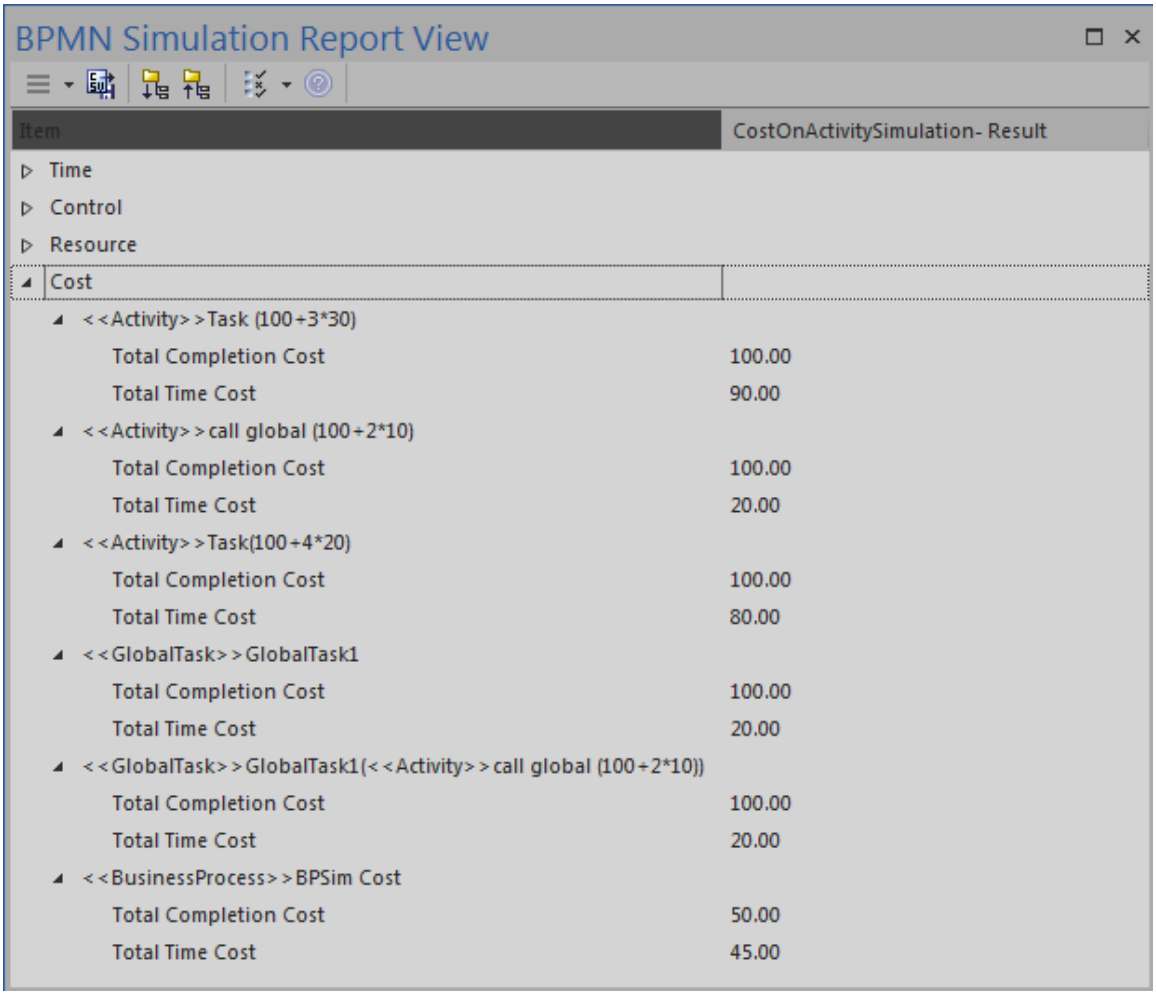
Create a Business Process Simulation configuration Artifact in the diagram, right-click on it and select the 'Configure BPSim' option. Set the configuration to link to the Package containing the BPMN model elements and configure these BPSim parameters as indicated.

Parameter	Settings
Scenario Parameters	1. Click on the BPSim Configuration Artifact and, for the Scenario Parameter 'Time Unit', click on the 'Value' drop-down arrow and select 'hours'. 2. In the 'Value' field for the 'Duration' parameter, set the value to '0001 00:00:00' (1 day). This Time Unit is used to calculate the Time Cost (Time cost = Unit Cost * Time), so make sure the Unit Cost is based on the correct Time Unit.
Control Parameters	1. On the diagram, click on *StartEvent1*. 2. Click on the *New Parameter* drop-down arrow and select 'Control'. 3. In the 'Parameter' field click on the drop-down arrow and select 'TriggerCount'. 4. In the 'Value' field type '1'.
Time Parameters	1. On the diagram click on *Task(100+4*20)*. 2. Click on the *New Parameter* drop-down arrow and select 'Time'. 3. Click on the 'Parameter' drop-down arrow and select 'ProcessingTime'. 4. In the 'Value' field set the value to '000:000:000 004:00:00' (4 hours). 5. Click on *Task(100+3*30)* on the diagram and repeat steps 2, 3 and 4, setting the 'Value' field to '000:000:000 003:00:00' (3 hours). 6. Click on *GlobalTask1* on the diagram and repeat steps 2, 3 and 4, setting the 'Value' field to '000:000:000 002:00:00' (2 hours).
Cost Parameters	1. On the diagram, click on *Task(100+4*20)*. 2. Click on the *New Parameter* drop-down arrow and select 'Cost'. 3. In the 'Parameter field click on the drop-down arrow and select, in turn:

	- 'FixedCost', then in the 'Value' field click on the button, select the 'Constant' tab and 'Floating', and in the 'Constant Floating' field type '100'; click on the OK button - 'UnitCost' - do the same, setting the 'Constant Floating' field to '20'. 4. On the diagram, click on *Task(100+3*30)* and repeat Steps 2 and 3, setting: - 'FixedCost' to '100 - 'UnitCost' to '30'. 5. On the diagram, click on *GlobalTask1* and repeat Steps 2 and 3, setting: - 'FixedCost' to '100 - 'UnitCost' to '10'. 6. On the diagram, click on *BPSim Cost* and repeat Steps 2 and 3, setting: - 'FixedCost' to '50' - 'UnitCost' to '5'.

Simulation

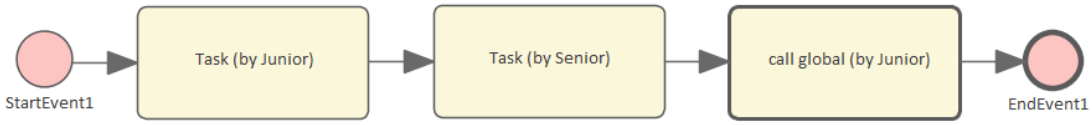
1. On the 'Configure BPSim' dialog, click on the 'Execute' tab.
2. Click on the button'.
3. When the simulation is complete, click on the 'Review' tab, and on the 'Standard Results Report' tab.
4. Filter the report by clicking on the button and selecting the 'Show only Non-Empty Items' option.



Analysis

Activity	Analysis
Task(100+4*20)	• Total Completion Cost is 100, matching the FixedCost (100) setting in BPSim • Total Time Cost is 80, calculated as ProcessingTime (4 hours) * UnitCost (20/hour)
Task(100+3*30)	• Total Completion Cost is 100, matching the FixedCost (100) setting in BPSim • Total Time Cost is 90, calculated as ProcessingTime (3 hours) * UnitCost (30/hour)
call global (100+2*10)	• Total Completion Cost is 100, matching the FixedCost (100) on *GlobalTask1* setting in BPSim • Total Time Cost is 20, calculated as ProcessingTime (2 hours) * UnitCost (10/hour) on *GlobalTask1*
BPSim Cost process	• Total Completion Cost is 50, matching the FixedCost (50) setting in BPSim • Total Time Cost is 45, calculated as Total ProcessingTime of all tasks (4 + 3 + 2 = 9 hours) * UnitCost (5/hour)

Set Cost Parameters on Resource



Create the BPMN Model (Resources)

1. In the Browser window create a *StartEvent1*, a *GlobalTask1*, two abstractTasks called *Task (by Junior)* and *Task (by Senior)*, and an *EndEvent1*.
2. Ctrl+drag the elements from the Browser window onto a diagram, pasting *GlobalTask1* as an Invocation (Call Activity) named *call global (by Junior).*
3. Connect the elements with Sequence Flows.
4. Create two BPMN2.0 Resource elements: *Junior Developer* and *Senior Developer.*

BPSim Configuration

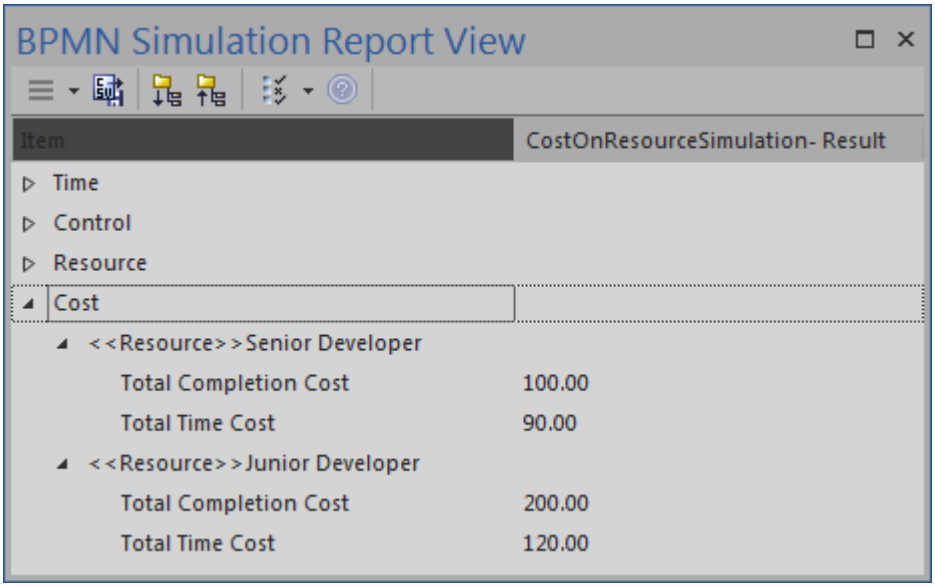
Create a Business Process Simulation configuration Artifact in the diagram, right-click on it and select the 'Configure BPSim' option, then set the configuration to link to the Package containing the BPMN model elements and configure these BPSim parameters as indicated.

Parameter	Setting
Scenario Parameters	1. Click on the BPSim Configuration Artifact and, for the Scenario Parameter 'Time Unit', click on the 'Value' drop-down arrow and select 'hours'. 2. In the 'Value' field for the 'Duration' parameter, set the value to '0001 00:00:00' (1 day). This Time Unit is used to calculate the Time Cost (Time cost = Unit Cost * Time), so make sure the Unit Cost is based on the correct Time Unit.
Control Parameters	1. On the diagram, click on *StartEvent1.* 2. Click on the *New Parameter* drop-down arrow and select 'Control'. 3. In the 'Parameter' field click on the drop-down arrow and select 'TriggerCount'. 4. In the 'Value' field type '1'.
Time Parameters	1. On the diagram click on *Task(by Junior).* 2. Click on the *New Parameter* drop-down arrow and select 'Time'. 3. Click on the 'Parameter' drop-down arrow and select 'ProcessingTime'. 4. In the 'Value' field set the value to '000:000:000 004:00:00' (4 hours). 5. Click on *Task (by Senior)* on the diagram and repeat steps 2, 3 and 4, setting the 'Value' field to '000:000:000 003:00:00' (3 hours). 6. Click on *GlobalTask1* on the diagram and repeat steps 2, 3 and 4, setting the 'Value' field to '000:000:000 002:00:00' (2 hours).
Resource Parameters	1. On the diagram, click on the *Junior Developer* Resource. 2. Click on the *New Parameter* drop-down arrow and select 'Resource'.

	3. Click on the 'Parameter' drop-down arrow and select 'Selection'. 4. In the 'Values' field, click on the button to open the 'Edit Resource Selection' dialog. 5. Click on 'Junior Developer', and on the Add Selection By Resource(s) button to move the selection to the 'Resource or Role' panel. 6. The 'Quantity Required' column defaults to '1'; overtype this value with '10'. 7. Click on the AND radio button to set the logical relationship; the final expression for Resource selection is composed and shown in the text field. 8. Click on the OK button to return to the Configure BPSim window, where the expression is shown in the 'Values' field. 9. Click on the *Senior Developer* Resource and repeat steps 2 through to 8, typing '5' in the 'Quantity Required' field.
Cost Parameters	1. On the diagram click on *Junior Developer*. 2. Click on the *New Parameter* drop-down arrow and select 'Resource'. 3. Click on the 'Parameter' drop-down arrow and select, in turn: - 'FixedCost', then in the 'Value' field click on the button, select the 'Constant' tab and 'Floating', then in the 'Constant Floating' field type '100' and in the 'CurrencyUnit' field type 'AUD'; click on the OK button - 'UnitCost' - do the same, setting the 'Constant Floating' field to '20'. 4. On the diagram click on *Senior Developer* and repeat steps 2 and 3, setting: - 'FixedCost' to '100' - 'UnitCost' to '30'.

Simulation

1. On the 'Configure BPSim' dialog, click on the 'Execute' tab.
2. Click on the button'.
3. When the simulation is complete, click on the 'Review' tab, and on the 'Standard Results Report' tab.
4. Filter the report by clicking on the button and selecting the 'Show only Non-Empty Items' option.



Analysis

Resource	Results
Junior Developer	• Total Completion Cost is '200', calculated as FixedCost (100) * number of Activities involved (2) • Total Time Cost is '120', calculated as ProcessingTime (4 + 2 = 6 hours) * UnitCost (20/hour)
Senior Developer	• Total Completion Cost is '100', calculated as FixedCost (100) * number of Activities involved (1) • Total Time Cost is '90', calculated as ProcessingTime (3 hours) * UnitCost (30/hour)

Export a BPSim Configuration

When you have defined a BPSim Configuration in a model, you can export it to an XMI file to be imported into other projects. The BPMN 2.0 model on which the configuration is based is also exported with the configuration. The model is bound to the appropriate BPSim Configuration when you import the XMI file into another project.

Access

Context Menu	On a diagram or in the Browser window, right-click on the Business Process Simulation Artifact | Export BPSim Configuration
Other	Toolbar of Configure BPSim window | Export icon

Publish Model Package

The process of exporting a BPSim configuration and its model uses the 'Publish Model Package' dialog for publishing a model to an XMI file.

Option	Description
Package	Defaults to the name of the Package containing the Business Process Simulation Artifact.
Filename	Type in or browse for (click on the icon) the file path and XML filename into which to export the model.
XML Type	Select 'BPMN 2.0 XML'.
Export	Click on this button to export the configuration and BPMN 2.0 model. The export is complete when a confirmation message displays in the 'Progress' field.
Format XML Output	Defaults to selected; leave selected.
View XML	If you want to examine the exported XML, click on this button.

Notes

- To import the model from XMI into a new project, select the target Package in the new project and select the 'Publish > Model Exchange > Import > Native File' or 'Import > XMI File' ribbon option