Tutorial AristaFlow BPM Suite

Next Generation Business Process Management

Automate business processes flexible

Orchestrate application process oriented

Realize human workflow without programming

Integrated process control seamlessly

Marlene-Dietrich-Str. 5
D-89231 Neu-Ulm

Telefon: +49 (0)731 9 85 88-600
Telefax: +49 (0)731 9 85 88

Modeling, validation, execution and monitoring of a time off request process using

ARISTAFLOW

Content and goals

This document gives you a quick start on using the tools and functions of the AristaFlow BPM-Suite by describing the development and implementation of a simple *request for time off* process step by step.

The goal of this overview tutorial is to give you an impression on how fast you can bring a business process alive with the AristaFlow BPM-Suite.

Please note that this tutorial can **not** substitute comprehensive trainings which are necessary to use the power of the AristaFlow BPM-Suite seriously. Advanced features like escalation, ad-hoc deviations, error handling, etc. and the usage of the powerful API to create extensions or even your own process centric applications are **not** covered by the tutorial.

Introduction

In normal situations the first step in the automation of business processes is to identify and analyze existing business processes using questionnaires or already existing process descriptions in order to define requirements for the process implementation. This is usually done on a semantically high level and may look like the following description for our running example: *Request for Time off Process*.

Process steps and involved participants in more detail:

1. Request for time off form is filled out by an employee.
2. The employee sends the request for time off to his / her supervisor.
3. The supervisor checks the request for time off and accepts or declines it.
4. The supervisor informs the employee if the request for time off has been accepted or declined.
5. If the request is declined the employee can withdraw his / her request or change it to submit it again.
6. If the request is accepted the supervisor informs the human resources department (HR)
7. The HR updates the employee's time off account.

After this rough specification we are ready to go on with the implementation of the process.

Implementation of the Request for Time off Process

In this section you will learn how to design a process model, how to create the corresponding process data / information, how to assign manual activities and automatic services and how to map your organizational model so that it can be used to specify the executing staff for each process steps.

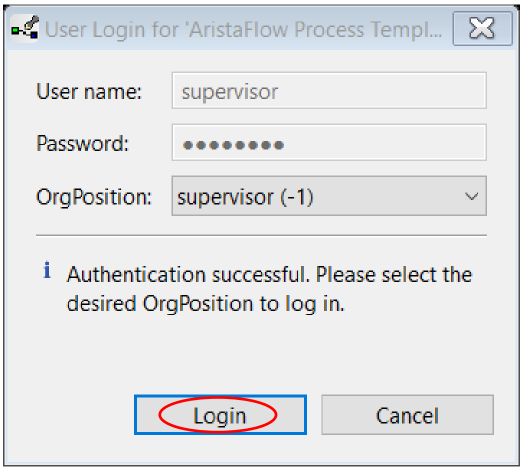
Hint:
The *AristaFlow Process Server* offers two possibilities for start-up. Using …\server\bin\server.bat the server for productive operations is started. Call …\aristaflow-server-transient\bin\server.bat to start a server without data persistence. The difference is that the transient version deletes the deployed process templates and running instances after shutdown, whereas the productive version keeps them permanently.
If you have installed the server as a Windows service, start it using the Windows service tool.

Note:
For the following steps it is required that the *AristaFlow Process Server* is running!

Designing the process

Start the AristaFlow *Process Template Editor* which you can find in your Windows start menu under … → *AristaFlow BPM Suite* → *Process Template Editor* or in your installation folder (…\ProcessTemplateEditor\ProcessTemplateEditor.exe).

Authenticate with a valid user name and password in the upcoming login dialogue (initially supervisor / password).



User name: supervisor

Password: ●●●●●●●●

OrgPosition: supervisor (-1)

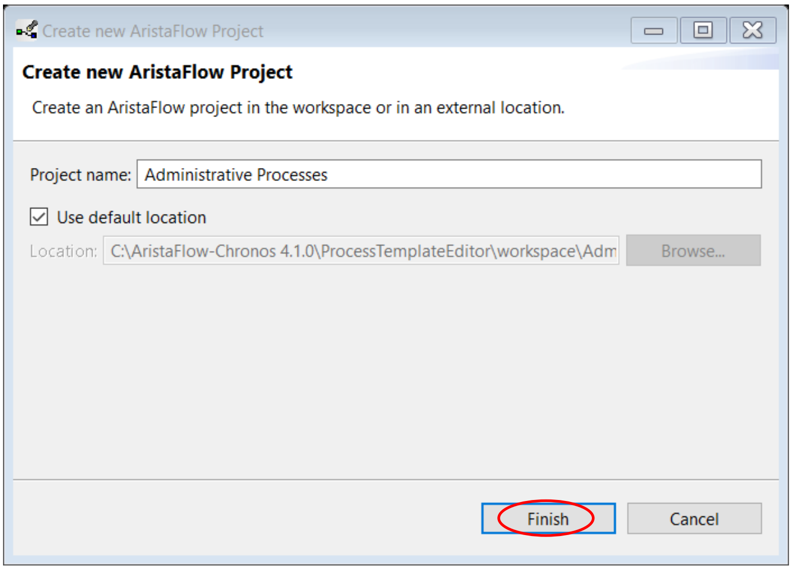
Authentication successful. Please select the desired OrgPosition to log in.

After login you enter the AristaFlow *Process Template Editor* where you can design arbitrary business processes of your company.

For the implementation of the *Request for Time off Process* the following steps are necessary:

1. Creation of a new process template

a. At first you have to create a new AristaFlow project. Right-click in the white area under the Tab *Local Projects* in the *AFResourceNavigator* view which you can find in the left upper corner of the *Process Template Editor*. Select *New* → *AristaFlow Project* and enter *Administrative Processes*. Close the dialogue.



Create new AristaFlow Project

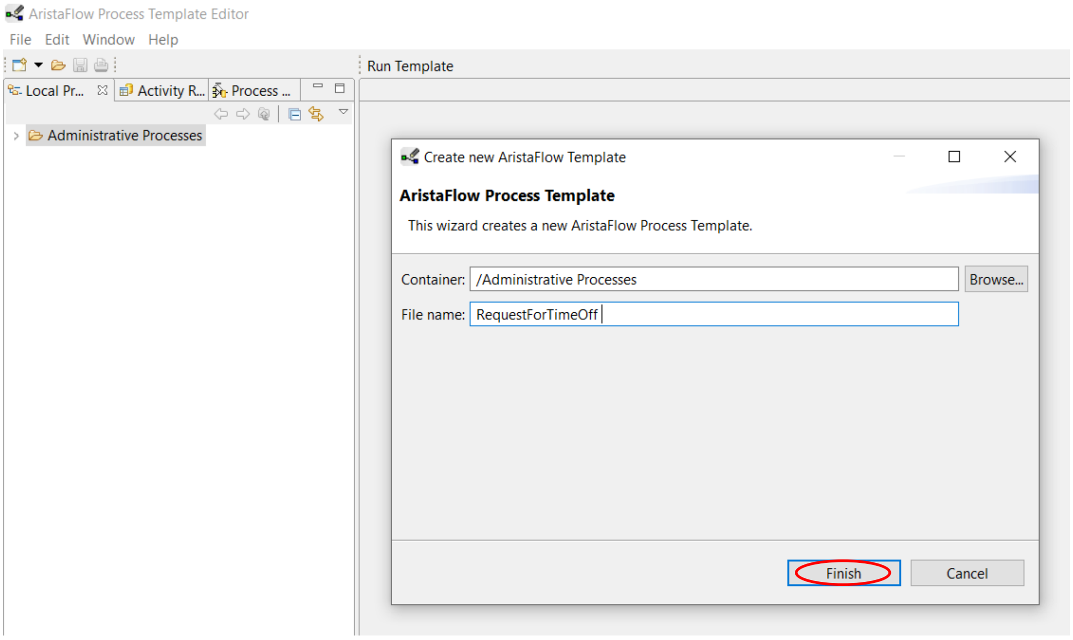
Create an AristaFlow project in the workspace or in an external location.

Project name: Administrative Processes

☑ Use default location

Location: C:\AristaFlow-Chronos 4.1.0\ProcessTemplateEditor\workspace\Adm

b. Now create the new process template. Right-click the newly created *Administrative Processes* project and select *New* → *AristaFlow Template*. Enter *RequestForTimeOff* in the File name input field of the dialogue and press finish.



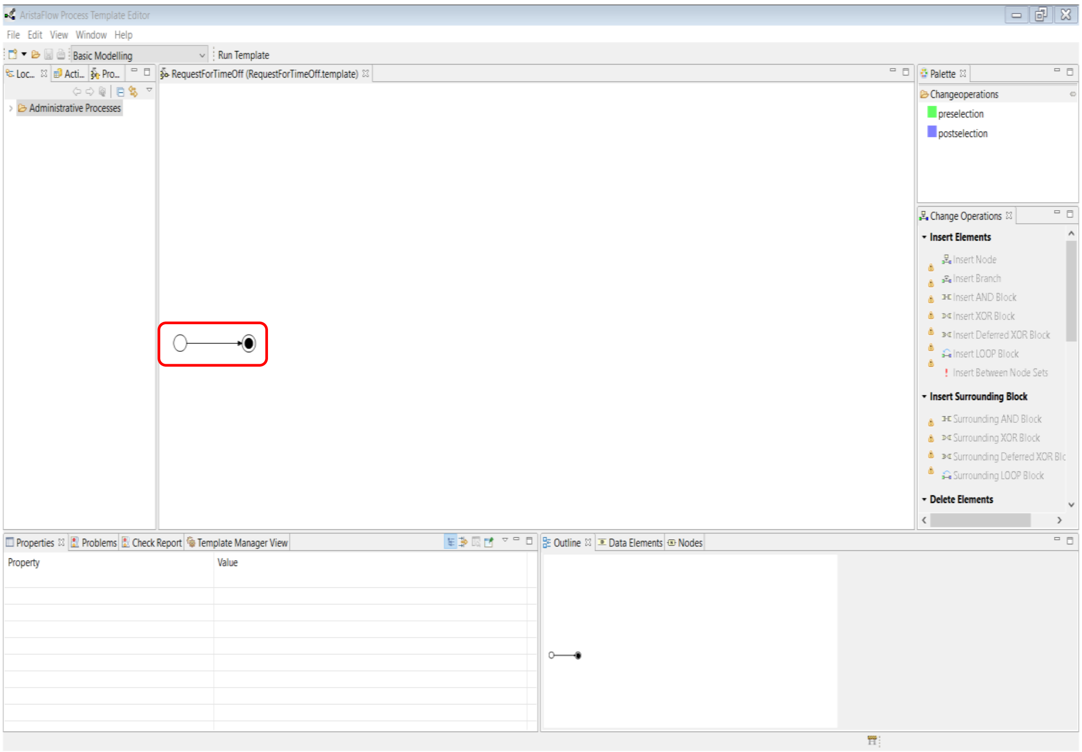
AristaFlow Process Template

This wizard creates a new AristaFlow Process Template.

Container: /Administrative Processes

File name: RequestForTimeOff

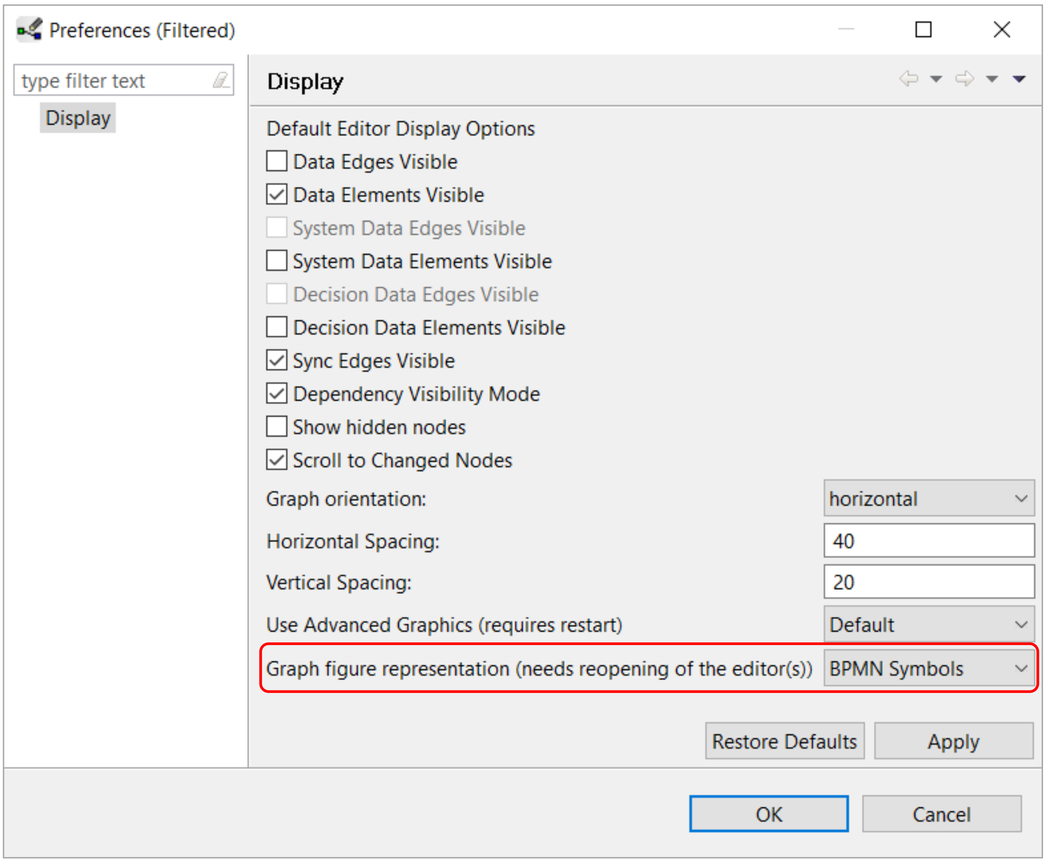
What you can see now in the working area of the *Process Template Editor* is the starting point for the design of every new business process: A single *Start* and its related *End* process step (in the following we sometimes use the simple term node as substitution for process steps).



Model the control flow of the process (process routing)

Hint:
Note that you may alternatively use *Adept symbols* in the process model. To do that, right-click the working area of the editor and select *Display options*. Then, choose *Adept Symbols* for *Graph figure representation* in the Display options window.

Evidence:
In this example *BPMN Symbols* were used.



Default Editor Display Options

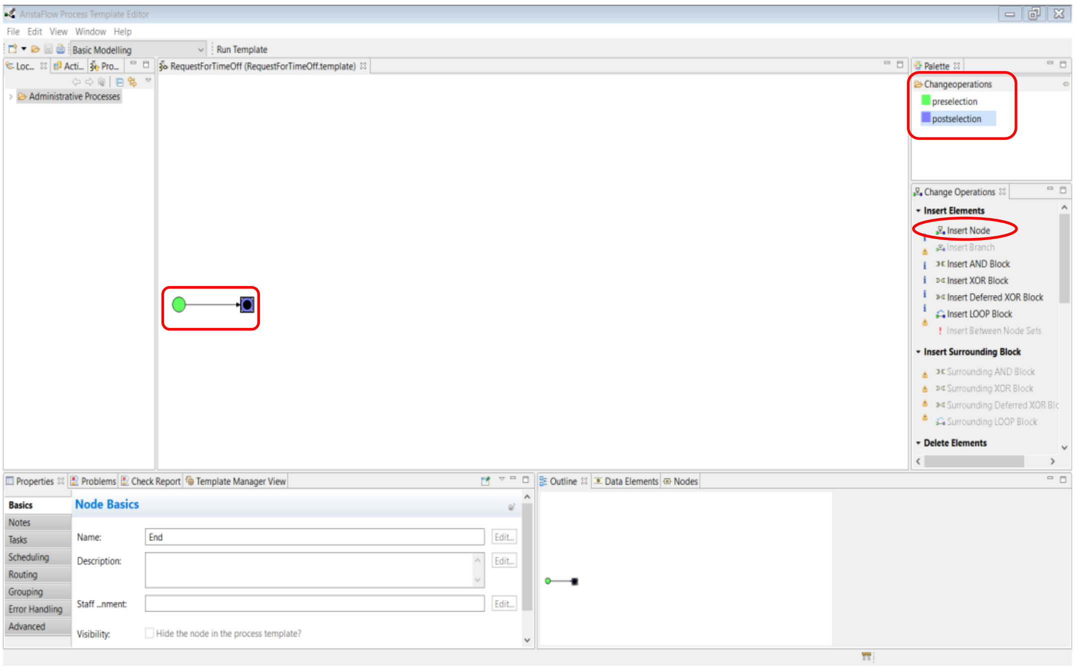
☐ Data Edges Visible
☑ Data Elements Visible
☐ System Data Edges Visible
☐ System Data Elements Visible
☐ Decision Data Edges Visible
☐ Decision Data Elements Visible
☑ Sync Edges Visible
☑ Dependency Visibility Mode
☐ Show hidden nodes
☑ Scroll to Changed Nodes

Graph orientation: horizontal
Horizontal Spacing: 40
Vertical Spacing: 20
Use Advanced Graphics (requires restart) Default
Graph figure representation (needs reopening of the editor(s)) BPMN Symbols

2. Model the control flow the process (process routing)

In contrast to all other process modeling tools you do not model your process graph only by drawing nodes and edges in the AristaFlow *Process Template Editor,* but also at the same time your model will be checked for correctness. The processes are designed by using a well defined set of process change operations like *Insert Node* or *Delete Node* to modify the process to your needs. By using these change operations it is guaranteed that the control flow model is valid at any point in time. Deadlocks, unreachable nodes or undefined process end points are automatically excluded. For business processes modeled with the AristaFlow *Process Template Editor* **Correctness by Construction** is guaranteed.

a. More precisely the control flow of a process model is designed by selecting a *pre-* and *postselection* (*preselection* and *postselection* in the upper right corner under Change Operations) followed by the selection of an enabled change operation on the right side of the editor. For example: Insertion of the first step "*Fill out request for time off*". Select the *Start node* in the *preselection* mode. The *Start node* turns green. Switch to *postselection* and select the *End node*. The End node turns blue.



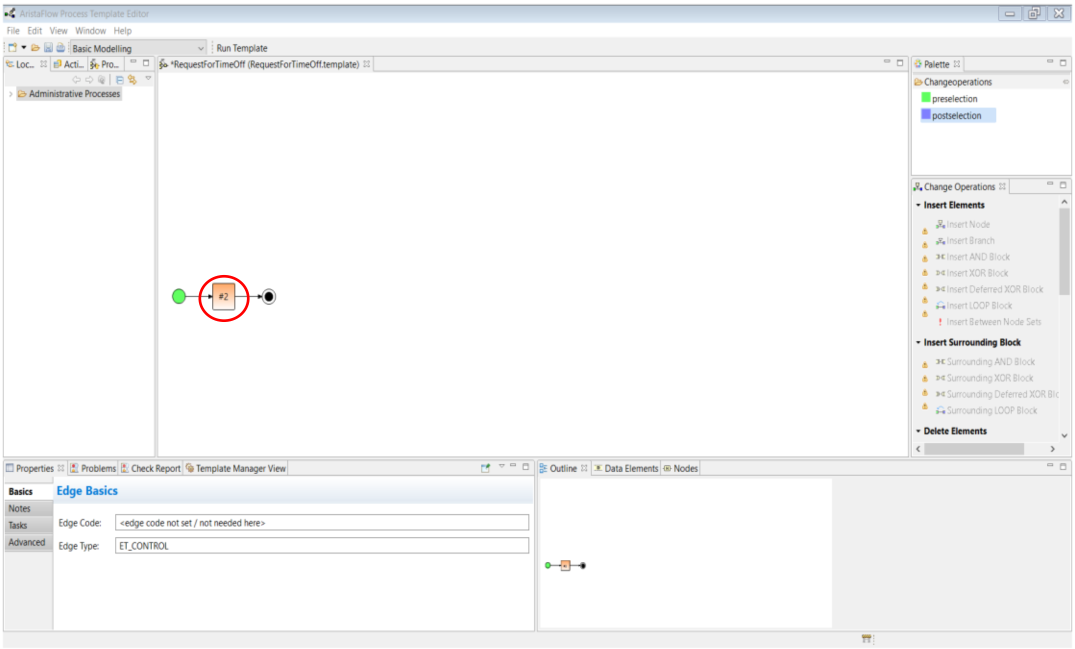
Name: End

Description:

Staff ...nment:

Visibility: □ Hide the node in the process template?

As a result of the selection exactly those change operations are enabled in the *Change Operations* View (on the right side) which can be applied in the given context without destroying the correctness of the process graph.
Select *Insert Node* and you can see that a new process step has been inserted automatically between the nodes of the *pre-* and *postselection*.



Edge Code: <edge code not set / not needed here>

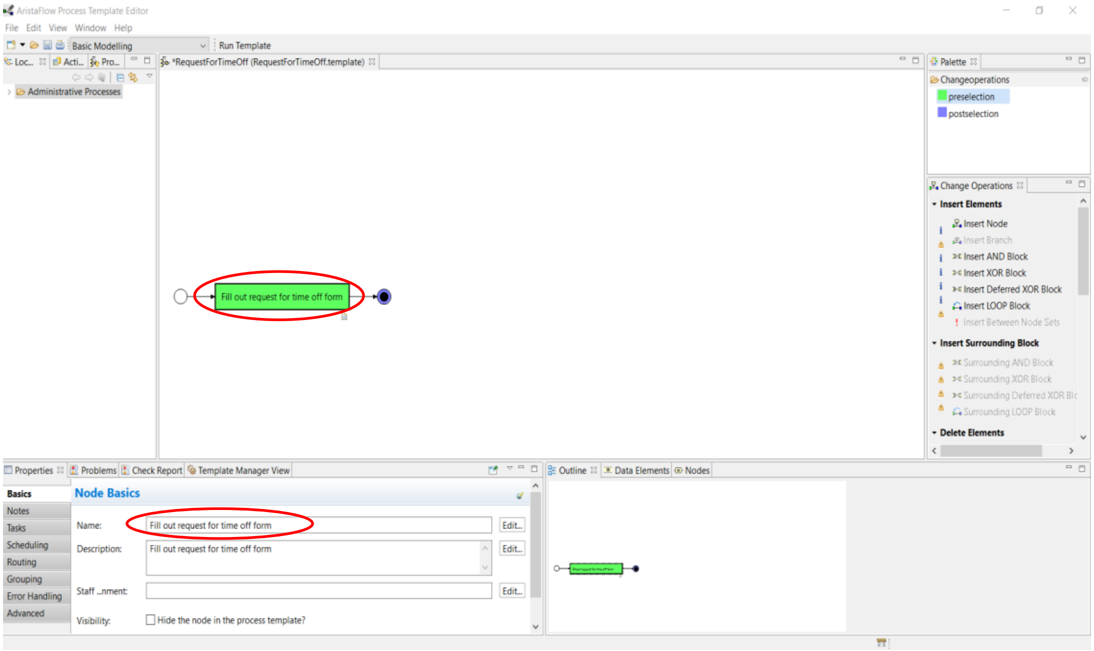
Edge Type: ET_CONTROL

Hint:
Instead of the explicit selection of the *pre-* and *postselection* in the upper right corner you can switch between the selections by using the middle mouse button.

Hint:
Tooltips on disabled change operations give information about the reason why there are not applicable in the given context.

b. To change the properties of a process step select the *Properties View* (bottom left). On the left hand side of the properties view you can see a couple of further tabs which groups the properties according to their functions like scheduling, error handling, routing, and so on.

For now the *Basics* tab is sufficient for our needs. Select the newly inserted node #2 and fill in the label of the node in the *Name* input field (“*Fill out request for time off form*”).



Node Basics

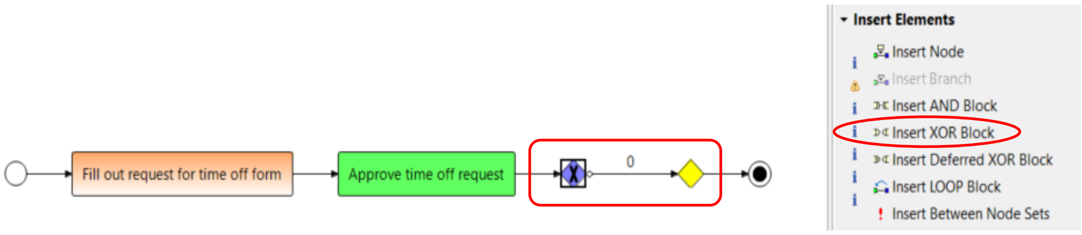
Name: Fill out request for time off form

Description: Fill out request for time off form

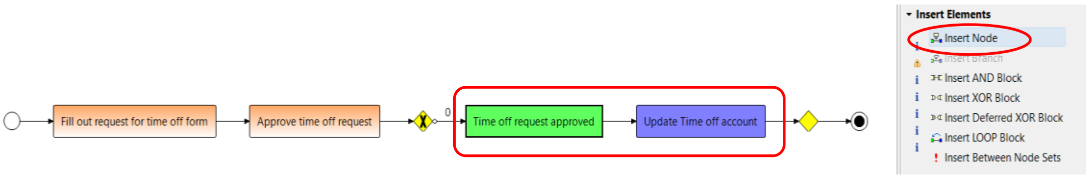
Staff ...nment:

Visibility: □ Hide the node in the process template?

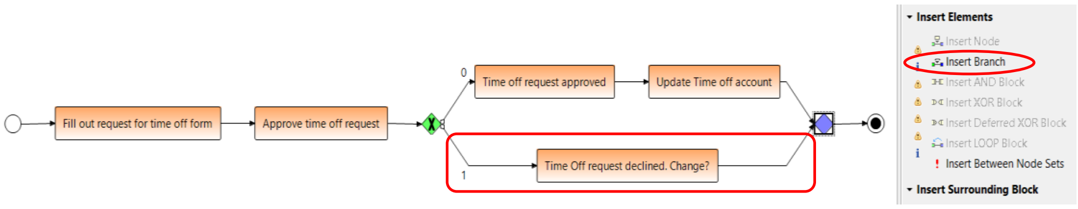
c. Design the next process step according to the explanations given in a. and b. and name it “Approve time off request”. After this node the request will be either approved or declined. This branch requires an XOR node. You can do this by “Insert XOR Block” in the Change Operations View and call it “approved/declined”.



d. On this branch you add the two nodes “Time off request approved” and “Update Time off account” according to a. and b.

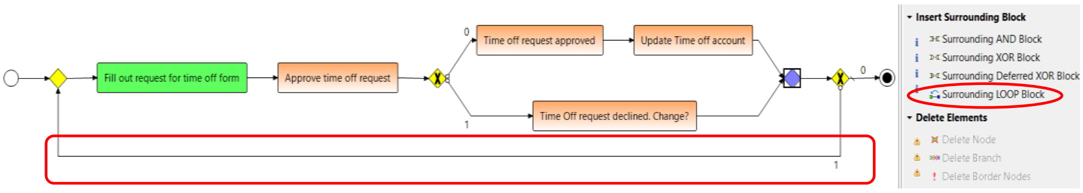


e. If the request is declined you need to add another branch by “Insert Branch” in the Change Operations View. On this branch you map the node “Time off request declined. Change?”.



f. In case of another request the workflow should go back to the start. You can do this Loop-Block with a “Surrounding LOOP-Block” over the first node “Fill out request for time off form” (as preselection) and the last XOR-Join node (as postselection). As name enter “Change time off request on decline?”.

The final result should look like this:



3. Model the data flow / the communication between the process steps.

In addition to the pure control flow of a business process we have to extend it with data or information which is exchanged between the single steps of a real life business process. In this chapter you learn how to create data elements / containers and how to link them with the process steps to obtain the process data flow. Note that the creation of data flows offers another unique feature of the AristaFlow *BPM-Suite*: The on the fly correctness checks of data flows which is a part of the **Correctness by Construction** paradigm. With this feature we can guarantee that all services, forms, etc. which are assigned to process steps are provided with the data values they need for a correct execution at runtime.

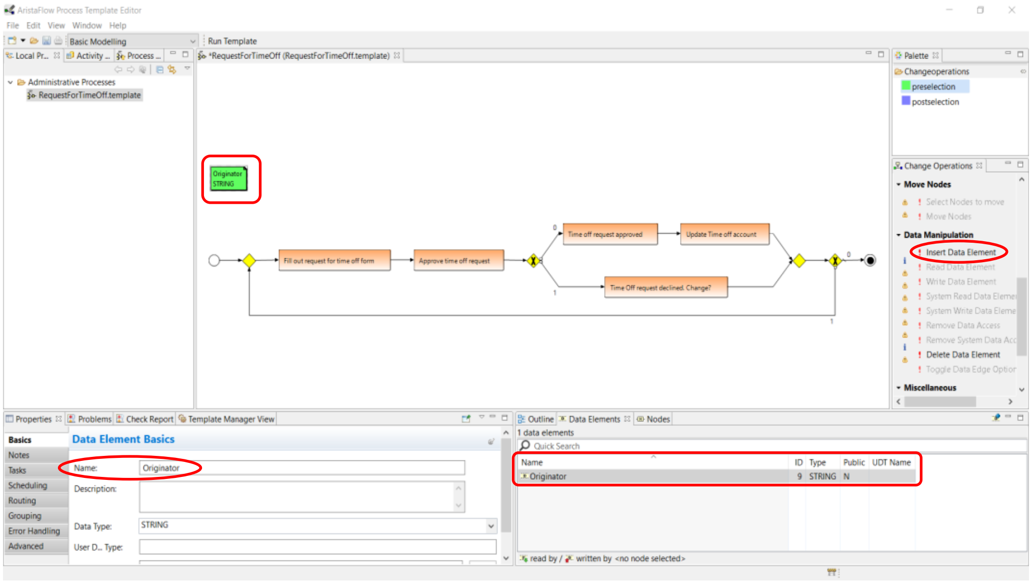
For our “*Request for Time off*” process we need the following data elements for a meaningful execution:

- Originator of the request (or in short: “*Originator*”)
- the beginning and the end of the time off period (“*Begin*”, “*End*”)
- a decision from the supervisor (“*Decision*”)
- a decision from the originator if he want to change its request if the initial request has been declined (“*Change*?”)

Create the data elements and the data flow as described below.

a. Create process data elements:
To create a new process data element select *Insert Data Element* in the *Change Operations* view. To change its properties select the data element and switch to the *Basics* tab of the *Properties View*. Alternatively switch to the *Data Element View* (in the lower right half of the editor) and select the data element value which you want to change.

i. For the process data element “*Originator*” create a new data element and set its *Data Type* to *String*.



Data Element Basics

Name: Originator

Description:

Data Type: STRING

User D... Type:

1 data elements

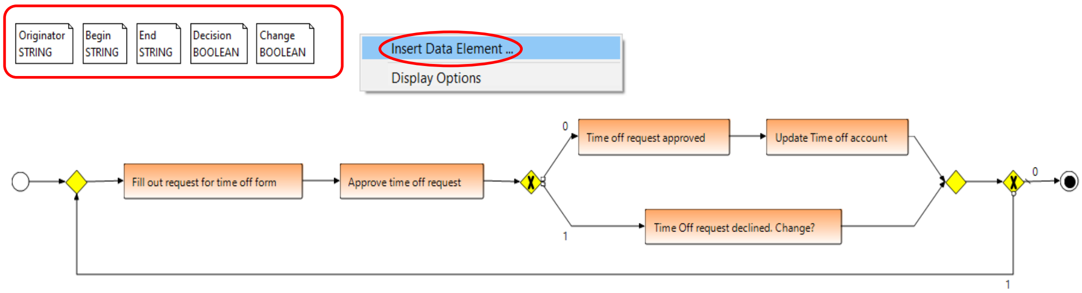
Name	ID	Type	Public	UDT Name
Originator	9	STRING	N	

read by / written by <no node selected>

ii. For “*Begin*” and “*End*” create new data elements and set their *Data Types* to *String* as well.

iii. For “*Decision*” and “*Change?*” create a new data elements and set their *Data Types* to *Boolean*.

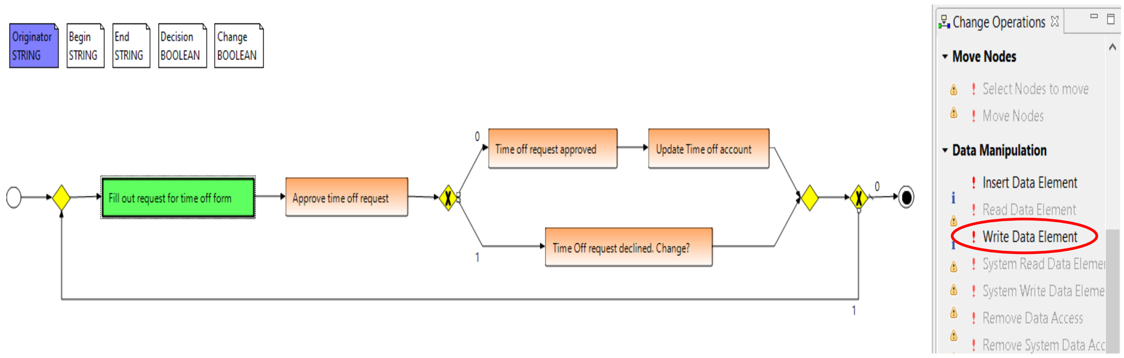
The result:



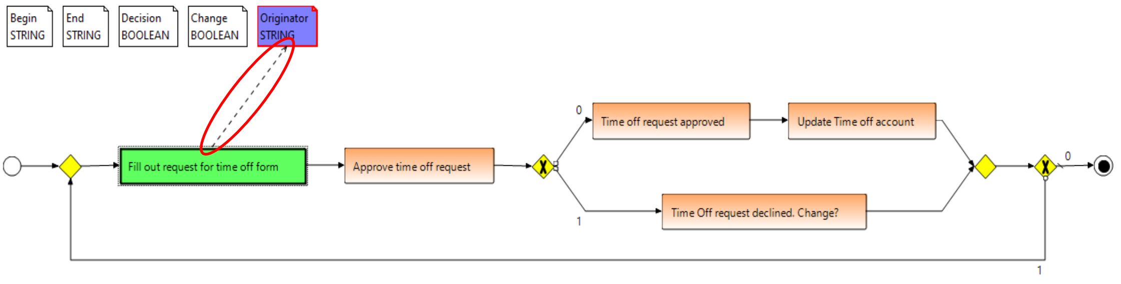
Model the data flow between the process steps:
To model the case that a process step delivers a data value you have to select the said process step as *preselection* and the data element where the data should be stored as *postselection*. Then select *Write Data Element* from the *Change Operations View*.

Hint:
To create a new data element simply right click over the graph into white area and select “Insert Data Element”. A Create new Data Element wizard opens in which you enter the name and choose the Type of the data element (see the last screenshot).

Example: Select the process step “*Fill out request for time off form*” as *preselection* and the data element “*Originator*” as *postselection* and push the *Write Data Element* button.

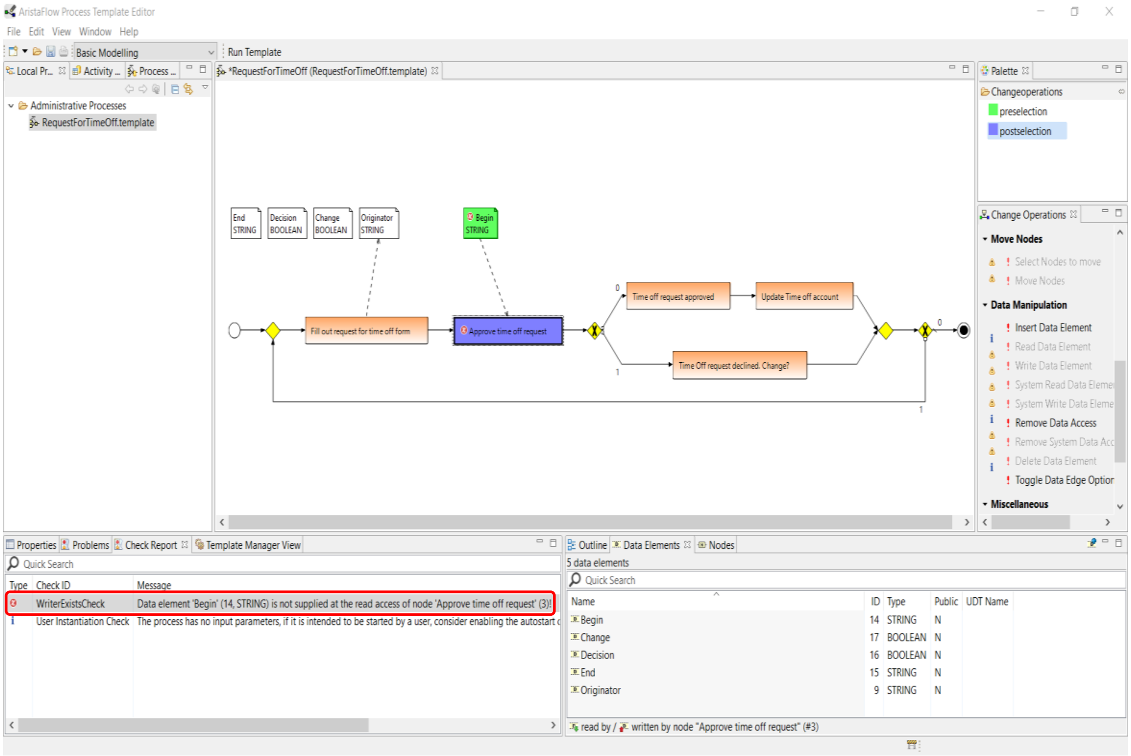


An arrow pointing towards the data element has been created.



Similar to the delivery of data values you model the case that a process step needs data values for its execution. In this case select the data element as *preselection* and the said process step as *postselection* followed by a *Read Data Element* change operation.

Example: Select the data element "*Begin*" as *preselection* and the process step "*Approve time off request*" as *postselection* and push the *Read Data Element* button.



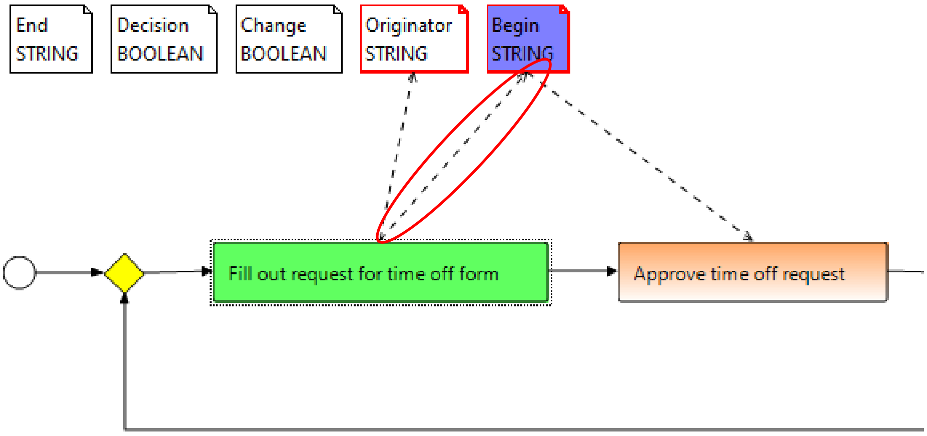
Type	Check ID	Message
⊗	WriterExistsCheck	Data element 'Begin' (14, STRING) is not supplied at the read access of node 'Approve time off request' (3)!
i	User Instantiation Check	The process has no input parameters, if it is intended to be started by a user, consider enabling the autostart c

5 data elements

Name	ID	Type	Public	UDT Name
Begin	14	STRING	N	
Change	17	BOOLEAN	N	
Decision	16	BOOLEAN	N	
End	15	STRING	N	
Originator	9	STRING	N	

read by / written by node "Approve time off request" (#3)

In the result you can see a red error symbol at the process step and the data element which is read by it. The *Check Report View* at the bottom informs us that no process step which is executed before the "*Approve time off request*" in the process flow delivers the "*Begin*" *data element value*. If we would try to execute this process it would definitely fail at runtime. The AristaFlow BPM-Suite checks those data flow errors on the fly no matter how complex the modeled processes are (including nested *XOR-Blocks*, *sub processes* etc.).
We fix this error by writing the *data element* from the "*Fill out request for time off form*".

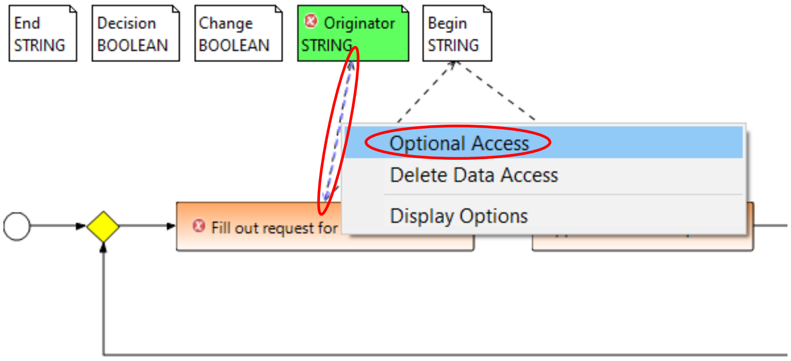


Hint:
To speed up the data flow modeling you can select more than a single data element or process step for read / write data edge operations by holding down the Ctrl. key during

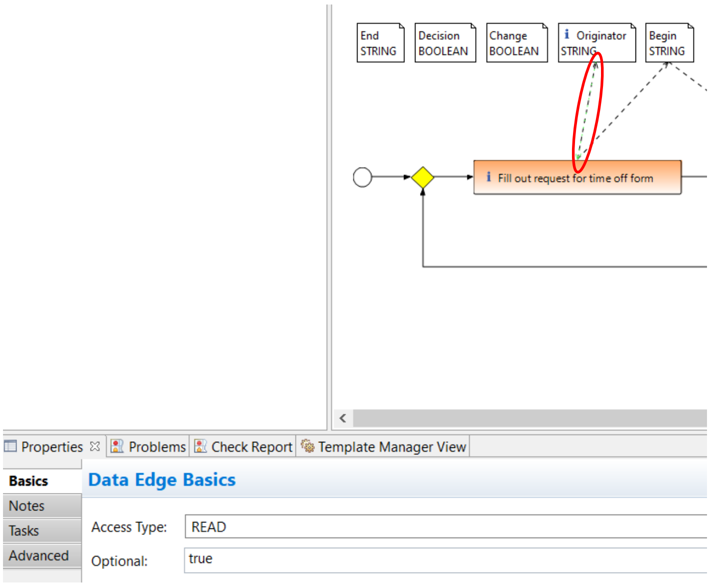
Complete the data flow in your process accordingly:

i. Process step "*Fill out request for time off form*"

Read (optional): "*Originator*", "*Begin*", *End*"
To read these three data elements is not essential but offers the possibility for the originator to see the data which he has filled in in the previous iteration. So it is much easier for him / her to correct the data.
To prevent data flow errors you have to mark these read data edges optional, because in the first iteration of the loop they do not contain values.
Select the read data edges one by one and choose "Optional Access" in the right-click menu.



Alternatively after selecting the read data edge you change its Optional attributes to "*true*" in the Properties View.



Access Type: READ

Optional: true

Write: "*Originator*", "*Begin*", "*End*"

i. Process step "*Time off request approved*"

Read: "*Originator*", "*Begin*", "*End*"

Write: "*Decision*"

ii. Process step "*approved / declined*"

Read: "*Decision*"

iii. Process step "*Update time off account*"

Read: "*Originator*", "*Begin*", "*End*"

iv. Process step "*Time off request declined. Change*?"

Write: "*Change?*"

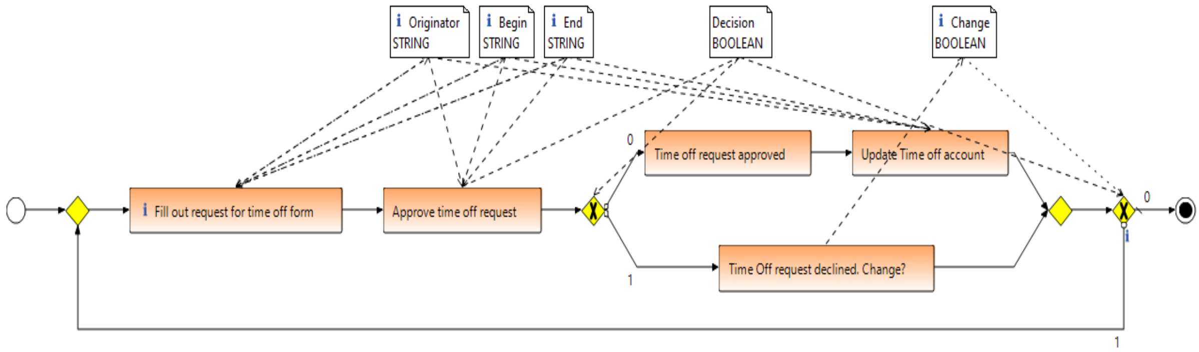
v. Process step "*Change time off request on decline?*"

Read: "*Decision*"

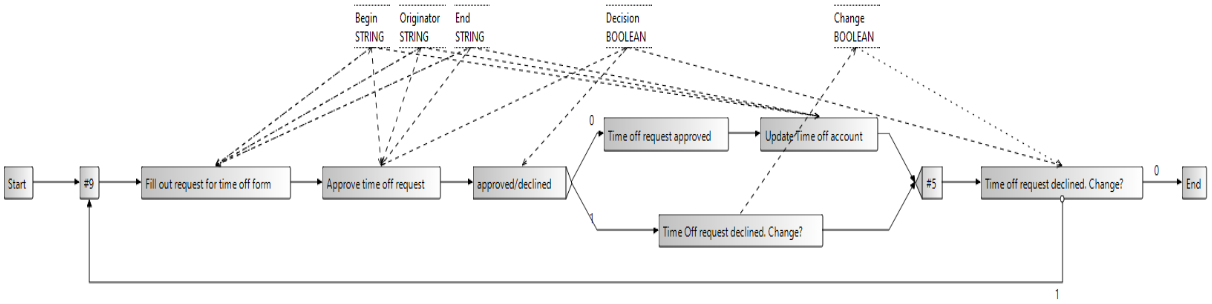
Read (optional): "*Change*"

In this case it is necessary to set the read data edge to optional because a -value is only delivered to "*Change?*" element if the request for time off was declined.

Result:



Result when using ADEPT symbols:



Hint:
For clarity reasons you can hide the data edges and / or the data elements. Right-click the working area of the editor and select *Display options*. Uncheck *Data Edges Visible* and / or *Data Elements Visible* in the upcoming dialogue. Furthermore you can check *Dependency Visibility Mode* to get a dynamic showing and hiding of the elements depending on the currently selected process step.

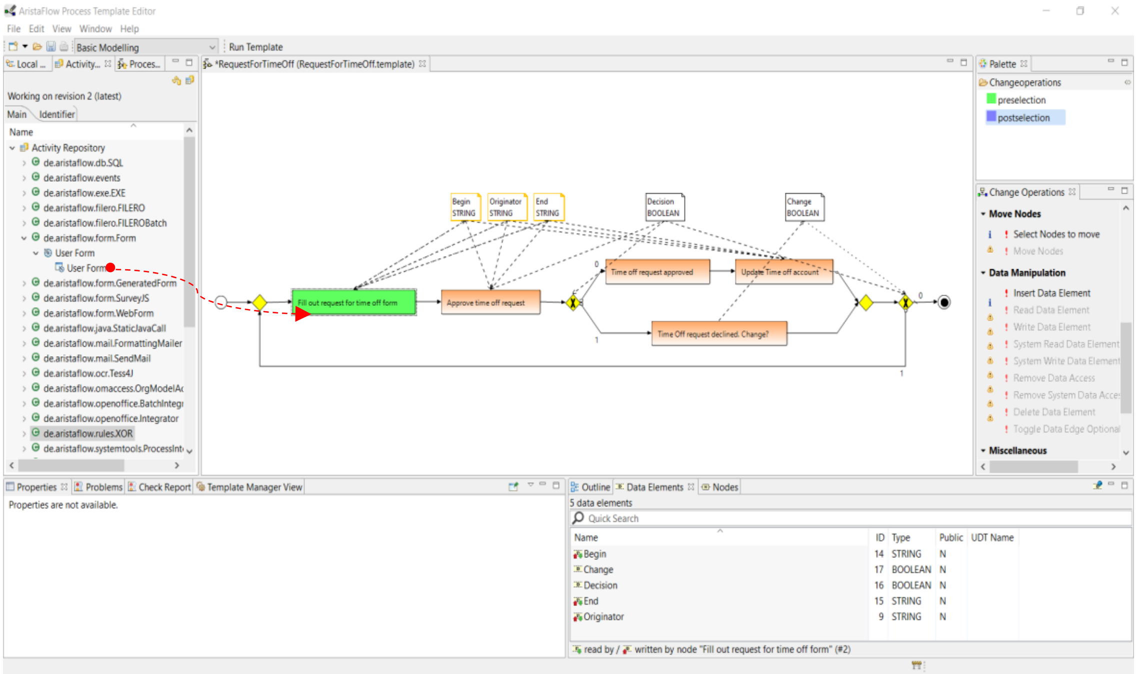
4. Assign activities / services to process steps

So far we modeled the control flow and the data flow of our business process. This can be very useful to get a first impression on how the automated process will look like and is a perfect technique to get an equal understanding on customer's requirements (*Rapid Prototyping*). For a productive operation of our process we have to assign activities / services to the process steps to animate them.

In AristaFlow we have a precise separation of process and application logic, so the activities / services are managed in a separate, versioned repository which is visualized in the *Activity Repository Browser* (the tab next to the *AFResourceNavigator* on the upper left side of the editor). Here you can find all components, functions and activities or more precisely activity templates which are available to be used in arbitrary business processes.

For our "*Request for time off*" process simply select an activity by expanding the parent component and its operation in the Activity Repository, drag it, move it to the process step where you want to assign it and drop it there.

Example: Expand *de.aristaflow.form.Form*. Expand *User Form* and select its child *User Form*. Keep the mouse button down, move the pointer to the "*Fill out request for time off*" process step and release the button (drag'n'drop as usual).


AristaFlow Process Template Editor
File Edit View Window Help
Basic Modelling
Run Template
Local ... Activity... Proces...
*RequestForTimeOff (RequestForTimeOff.template)
Palette
Changeoperations
preselection
postselection
Working on revision 2 (latest)
Main Identifier
Name
Activity Repository
de.aristaflow.db.SQL
de.aristaflow.events
de.aristaflow.exe.EXE
de.aristaflow.filero.FILERO
de.aristaflow.filero.FILEROBatch
de.aristaflow.form.Form
User Form
User Form
de.aristaflow.form.GeneratedForm
de.aristaflow.form.SurveyJS
de.aristaflow.form.WebForm
de.aristaflow.java.StaticJavaCall
de.aristaflow.mail.FormattingMailer
de.aristaflow.mail.SendMail
de.aristaflow.ocr.Tess4J
de.aristaflow.omaccess.OrgModelAc
de.aristaflow.openoffice.BatchIntegr
de.aristaflow.openoffice.Integrator
de.aristaflow.rules.XOR
de.aristaflow.systemtools.ProcessInt
Begin STRING
Originator STRING
End STRING
Decision BOOLEAN
Change BOOLEAN
Fill out request for time off form
Approve time off request
Time off request approved
Update Time off account
Time Off request declined. Change?
0
1
0
1
Change Operations
Move Nodes
Select Nodes to move
Move Nodes
Data Manipulation
Insert Data Element
Read Data Element
Write Data Element
System Read Data Element
System Write Data Element
Remove Data Access
Remove System Data Acces
Delete Data Element
Toggle Data Edge Optional
Miscellaneous
Properties Problems Check Report Template Manager View
Properties are not available.
Outline Data Elements Nodes

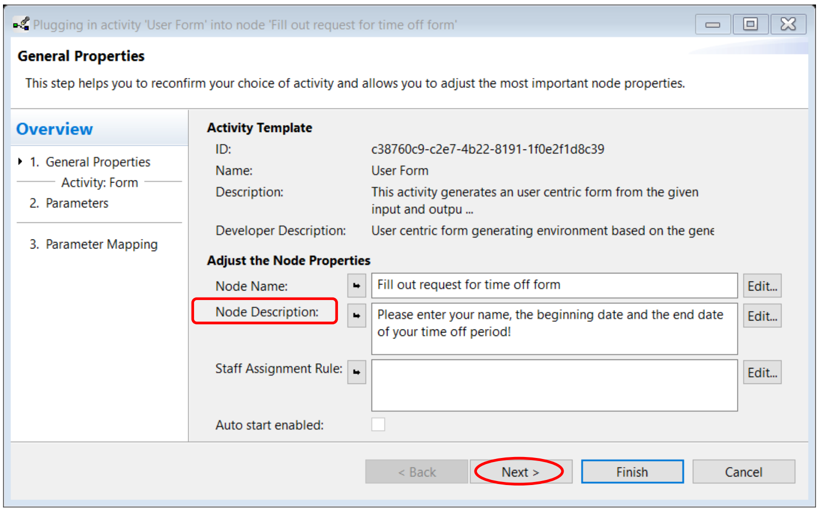
5 data elements

Quick Search

Name	ID	Type	Public	UDT Name
Begin	14	STRING	N	
Change	17	BOOLEAN	N	
Decision	16	BOOLEAN	N	
End	15	STRING	N	
Originator	9	STRING	N	

read by / written by node "Fill out request for time off form" (#2)

An activity template specific wizard opens which guides you through the following steps.



General Properties

This step helps you to reconfirm your choice of activity and allows you to adjust the most important node properties.

Activity Template

ID: c38760c9-c2e7-4b22-8191-1f0e2f1d8c39

Name: User Form

Description: This activity generates an user centric form from the given input and outpu ...

Developer Description: User centric form generating environment based on the gene

Adjust the Node Properties

Node Name: Fill out request for time off form Edit...

Node Description: Please enter your name, the beginning date and the end date of your time off period! Edit...

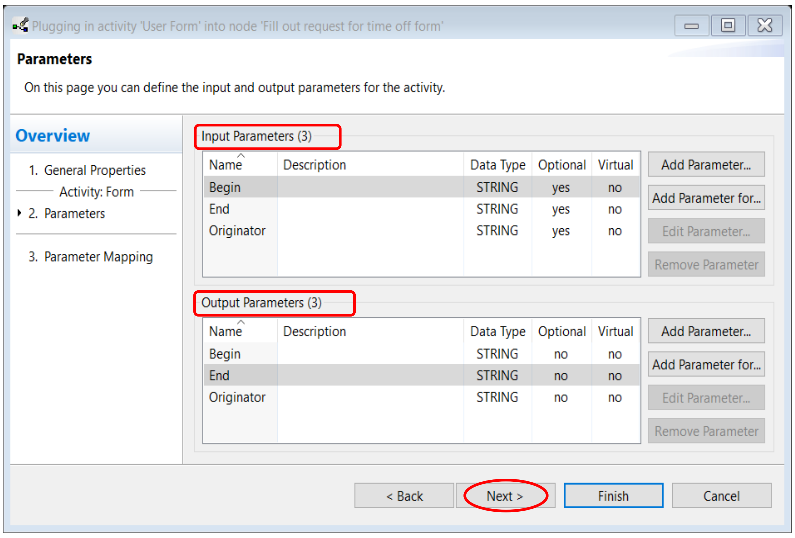
Staff Assignment Rule: Edit...

Auto start enabled: □

You can enter a description in the *Node Description* field to specify the process step more precisely. In case of the "*User Form*" activity this description is also shown as header for the form at runtime. Enter something like "*Please enter your name, the beginning date and the end date of your time off period*". Furthermore you can assign staff which is allowed to execute this process step, but we will carry about that later.

Hint:
Some of the activity templates already offer suggestions for Node *Name* and Node *Description*. Press the arrow button in front of the input fields to take them over.

Go on with *Next* > to the parameters page. Here you can manage all *input* and *output parameters* of the process step.


Plugging in activity 'User Form' into node 'Fill out request for time off form'

Parameters

On this page you can define the input and output parameters for the activity.

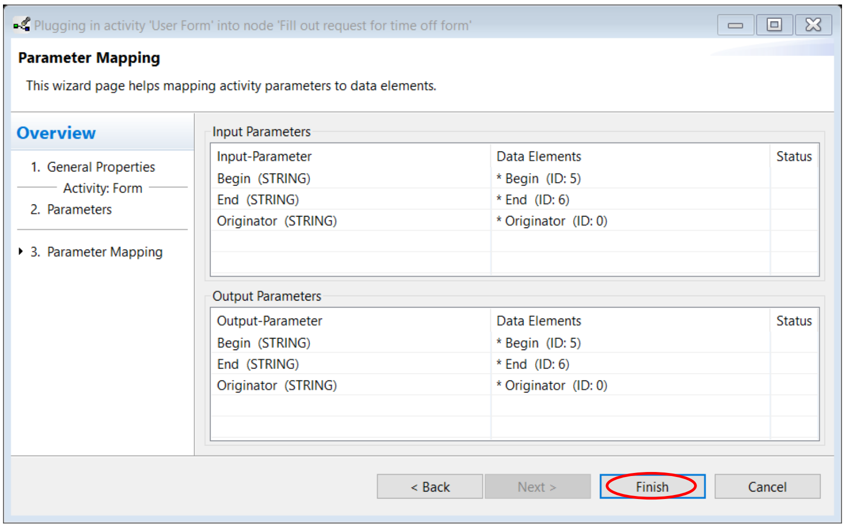
Input Parameters (3)

Name	Description	Data Type	Optional	Virtual
Begin		STRING	yes	no
End		STRING	yes	no
Originator		STRING	yes	no

Output Parameters (3)

Name	Description	Data Type	Optional	Virtual
Begin		STRING	no	no
End		STRING	no	no
Originator		STRING	no	no

The next wizard page offers a suggestion on how to map the input and output parameters of the process step to the process data elements. Thereby only those data elements are selectable in the drop down boxes which are compliant to the related parameter of the activity (**Correctness by Construction** paradigm). The asterisk signals that the selected data element is already connected to the process step via data edge. In our example the suggested mapping already matches so we can finish the wizard without further changes.


Plugging in activity 'User Form' into node 'Fill out request for time off form'

Parameter Mapping

This wizard page helps mapping activity parameters to data elements.

Input Parameters

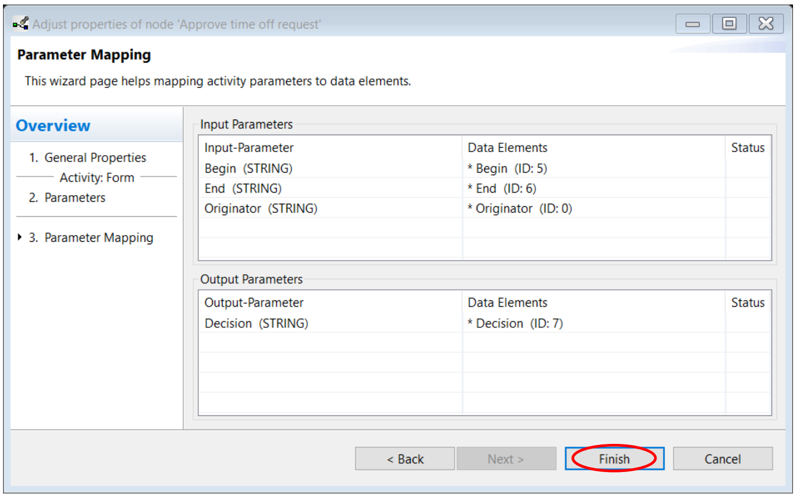
Input-Parameter	Data Elements	Status
Begin (STRING)	* Begin (ID: 5)	
End (STRING)	* End (ID: 6)	
Originator (STRING)	* Originator (ID: 0)	

Output Parameters

Output-Parameter	Data Elements	Status
Begin (STRING)	* Begin (ID: 5)	
End (STRING)	* End (ID: 6)	
Originator (STRING)	* Originator (ID: 0)	

Assign the other activities to the process steps:

a. Process step "*Approve time off request*": *de.aristaflow.form.Form* → *User Form* → *User Form*. Enter a description like "*Please approve / decline the request for time off*", check the mapping (should match already) and finish the wizard.



Parameter Mapping

This wizard page helps mapping activity parameters to data elements.

Input Parameters

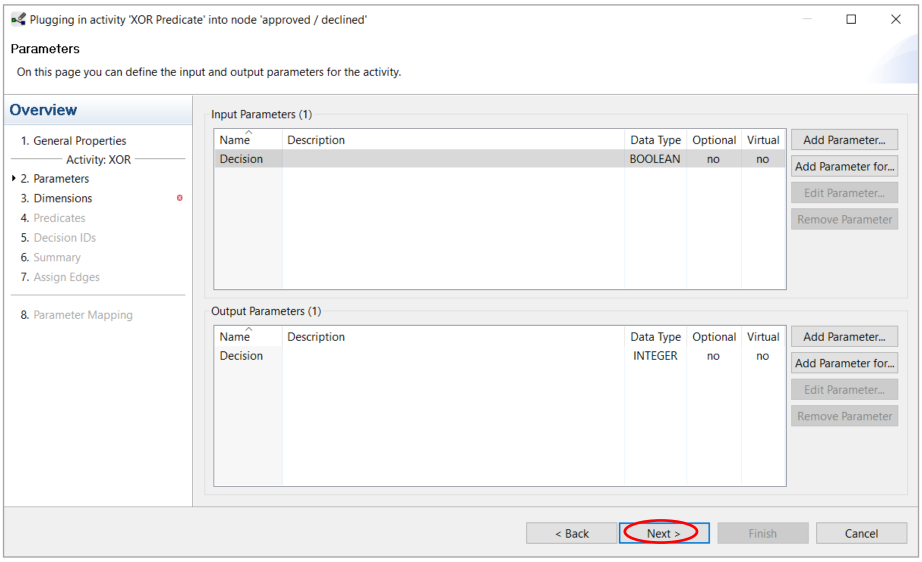
Input-Parameter	Data Elements	Status
Begin (STRING)	* Begin (ID: 5)	
End (STRING)	* End (ID: 6)	
Originator (STRING)	* Originator (ID: 0)	

Output Parameters

Output-Parameter	Data Elements	Status
Decision (STRING)	* Decision (ID: 7)	

b. Process step "*approved / declined*": *de.aristaflow.rules.XOR* → *XOR Predicate* → *XOR Predicate*. This is a special activity for defining branch decisions of simple to arbitrary complex type (*business rules*). The special feature of this AristaFlow activity is the check that all possible input value combinations used for the decision are mapped to an outgoing branch. Situations where you make a decision like "> 1000" then take branch X and "< 1000" then take branch Y and you forgot "= 1000" are checked automatically at buildtime and is shown to the user. The situation described in the example is simple, but the check also applies for arbitrary complex decisions with many parameters involved.

We turn back to the configuration of the decision activity. We switch to the second page with is equal to the parameter page of the *User Form* activity. In this case the listed *Input Parameters* are the parameters which are used in our branch decision (business rule). Since the process step already contains ingoing data edges compliant parameter were created automatically. The *Output Parameter* "*Decision*" is a system parameter and is always created. Do not change it.



Plugging in activity 'XOR Predicate' into node 'approved / declined'

Parameters

On this page you can define the input and output parameters for the activity.

Input Parameters (1)

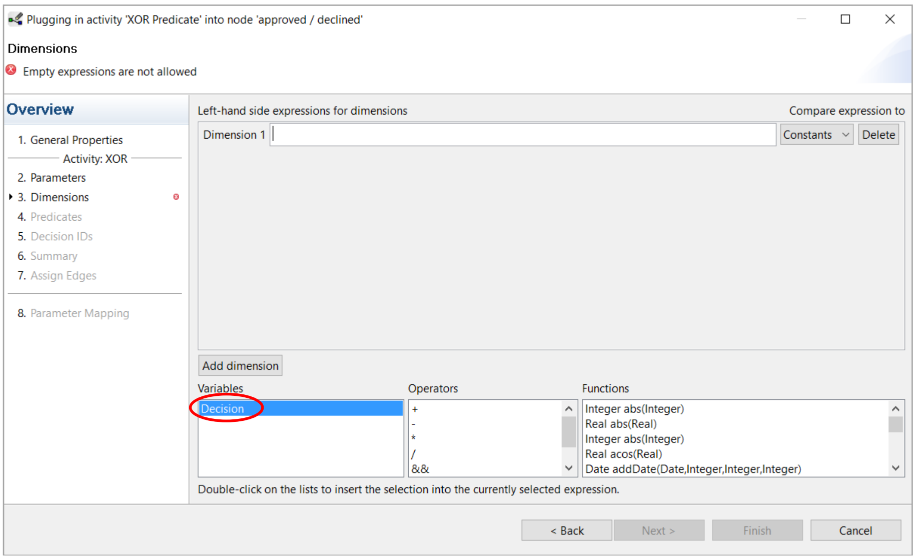
Name	Description	Data Type	Optional	Virtual
Decision		BOOLEAN	no	no

Output Parameters (1)

Name	Description	Data Type	Optional	Virtual
Decision		INTEGER	no	no

On the next wizard page we define the dimensions of the decision. For a better understanding of the term dimension you can consider a solution space which covers all values for a certain type. For example *Boolean* would have a solution space of *true* and *false* while *Integer* would have one from -2^{31} to 2^{31}-1. The challenge for the decision activity is now to guarantee that the whole solution space is covered if the user divides it by defining decision statements like value “*< 1000*” for *Integer* or value “*= TRUE*” for *Boolean*.

For our simple example with a single *Boolean* value we only have to **double**-click *Decision* and switch to the next page.



Plugging in activity 'XOR Predicate' into node 'approved / declined'

Dimensions

Empty expressions are not allowed

Overview

1. General Properties
   Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

Left-hand side expressions for dimensions | Compare expression to

Dimension 1 | Constants | Delete

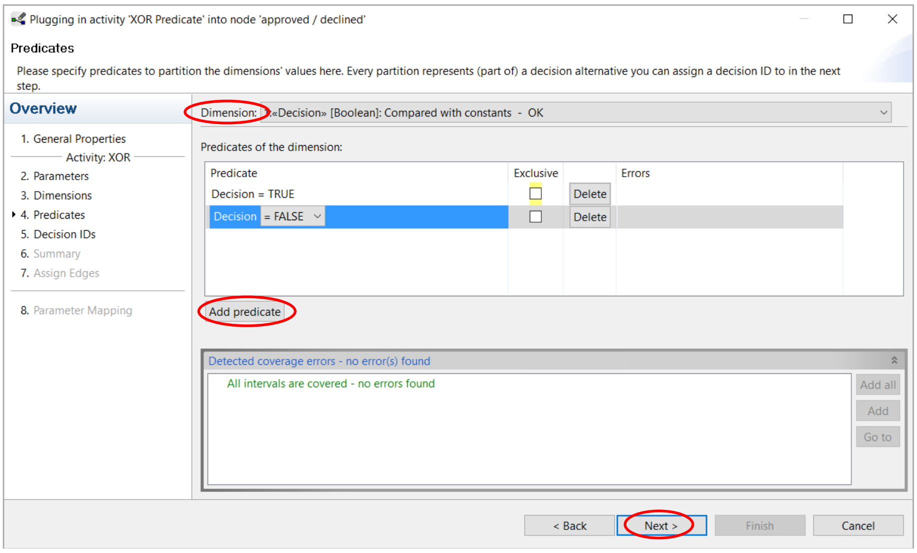
Variables: Decision

Operators: + - * / &&

Functions: Integer abs(Integer), Real abs(Real), Integer abs(Integer), Real acos(Real), Date addDate(Date,Integer,Integer,Integer)

Double-click on the lists to insert the selection into the currently selected expression.

On this page the predicates and the values for the dimensions are specified. As you can see from the following screenshot the *XOR wizard* calculates the uncovered values automatically. In our example only the case “*Decision = TRUE*” is covered already. To cover the value “*Decision = FALSE*” click the *Add Predicate* button and select *false* in the drop down box. Alternatively you can select “*Add all*” to complete the decision values for our *input parameter*.



Plugging in activity 'XOR Predicate' into node 'approved / declined'

Predicates

Please specify predicates to partition the dimensions' values here. Every partition represents (part of) a decision alternative you can assign a decision ID to in the next step.

Overview

1. General Properties
   Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

Dimension: «Decision» [Boolean]: Compared with constants - OK

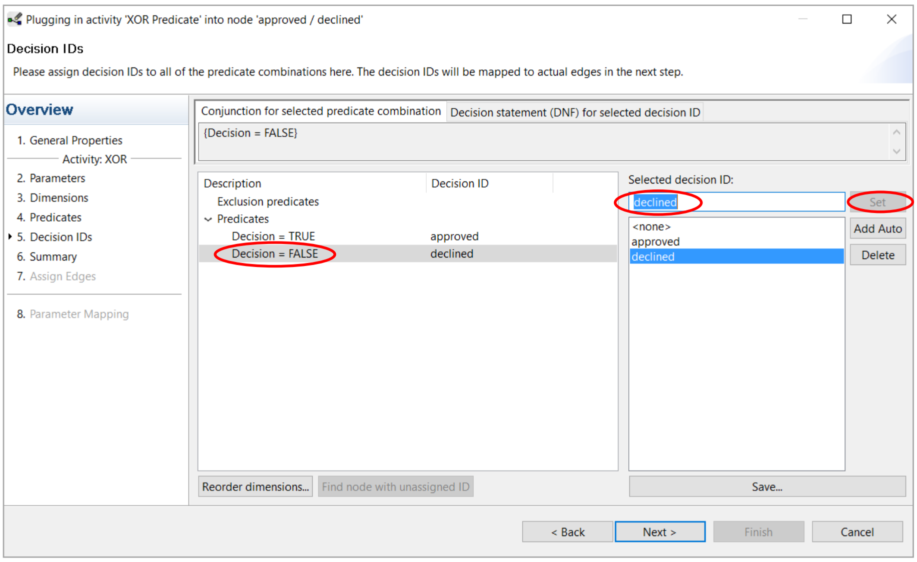
Predicates of the dimension:

Predicate	Exclusive		Errors
Decision = TRUE	□	Delete	
Decision = FALSE	□	Delete	

Detected coverage errors - no error(s) found

All intervals are covered - no errors found

On the next wizard page you have to assign decision IDs to the predicates. Select “*Decision = TRUE*” and click the *Add Auto* button to create an ID automatically or enter your own ID in the input field and choose *Set*.


Plugging in activity 'XOR Predicate' into node 'approved / declined'

Decision IDs

Please assign decision IDs to all of the predicate combinations here. The decision IDs will be mapped to actual edges in the next step.

Overview

1. General Properties
Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

Conjunction for selected predicate combination | Decision statement (DNF) for selected decision ID

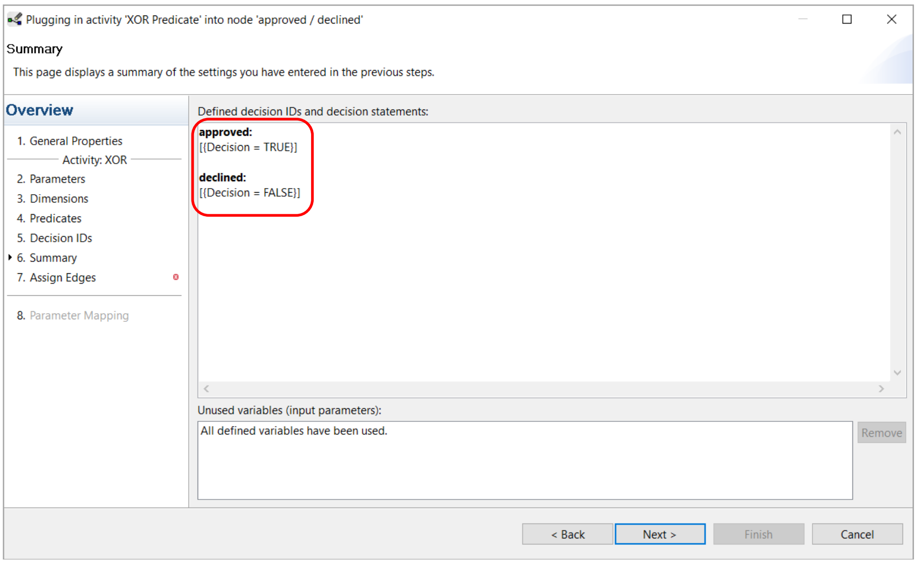
{Decision = FALSE}

Description	Decision ID
Exclusion predicates	
Predicates	
Decision = TRUE	approved
Decision = FALSE	declined

Selected decision ID: declined

<none>
approved
declined

We enter our own ID and call it “*approved*“. For “*Decision = FALSE*” we enter “*declined*” and switch to the summary page for an overview on our branch decision.


Plugging in activity 'XOR Predicate' into node 'approved / declined'

Summary

This page displays a summary of the settings you have entered in the previous steps.

Overview

1. General Properties
Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

Defined decision IDs and decision statements:

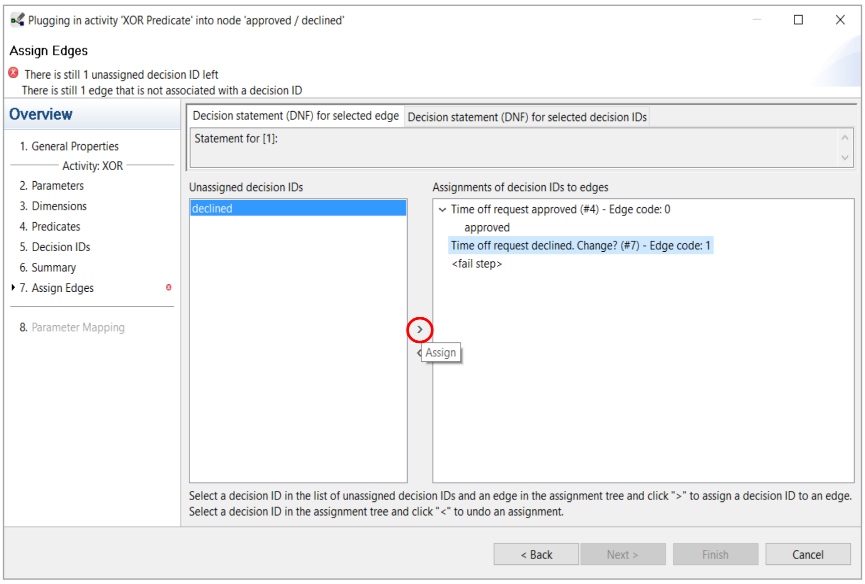
approved:
[{Decision = TRUE}]

declined:
[{Decision = FALSE}]

Unused variables (input parameters):

All defined variables have been used.

In the next step the decision IDs are connected to the outgoing branches of the decision split node. Select the decision ID “*approved*” and the branch with the node “*Time off request approved*” and connect them by clicking the arrow button. Connect the “*declined*” ID with “*Time off request declined. Change?*“ accordingly.



Plugging in activity 'XOR Predicate' into node 'approved / declined'

Assign Edges

There is still 1 unassigned decision ID left
There is still 1 edge that is not associated with a decision ID

Overview

1. General Properties
Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

Decision statement (DNF) for selected edge | Decision statement (DNF) for selected decision IDs

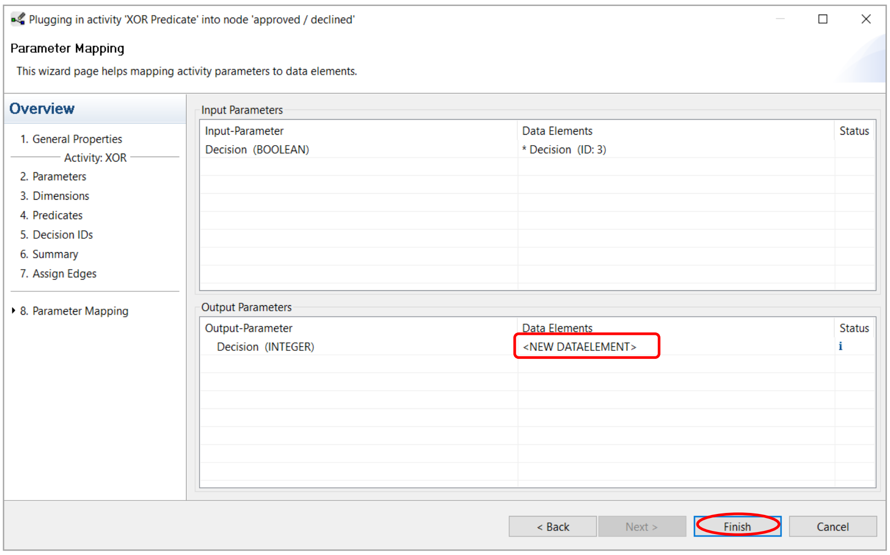
Statement for [1]:

Unassigned decision IDs
declined

Assignments of decision IDs to edges
Time off request approved (#4) - Edge code: 0
approved
Time off request declined. Change? (#7) - Edge code: 1
<fail step>

Select a decision ID in the list of unassigned decision IDs and an edge in the assignment tree and click ">" to assign a decision ID to an edge.
Select a decision ID in the assignment tree and click "<" to undo an assignment.

To finish the decision wizard the parameters have to be mapped to the data flow using the already known mapping dialogue. The mapping for the input parameter “*Decision*” should be valid already. For the system output parameter “*Decision*” select *<NEW DATAELEMENT>*.



Plugging in activity 'XOR Predicate' into node 'approved / declined'

Parameter Mapping

This wizard page helps mapping activity parameters to data elements.

Overview

1. General Properties
Activity: XOR
2. Parameters
3. Dimensions
4. Predicates
5. Decision IDs
6. Summary
7. Assign Edges
8. Parameter Mapping

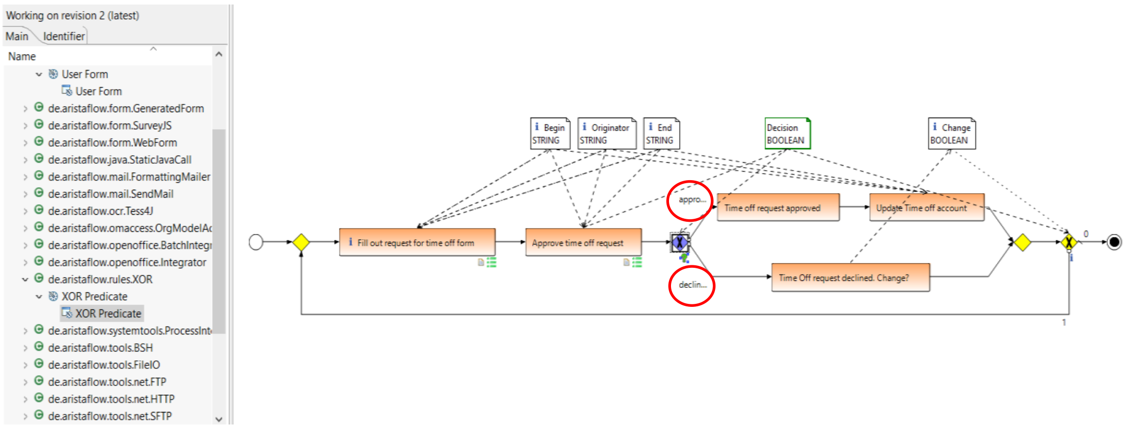
Input Parameters

Input-Parameter	Data Elements	Status
Decision (BOOLEAN)	* Decision (ID: 3)	

Output Parameters

Output-Parameter	Data Elements	Status
Decision (INTEGER)	<NEW DATAELEMENT>	i

Close the wizard with the *Finish* button. Now you can see on the decision activity’s outgoing control flow edges which branch is selected in accordance to the input value.

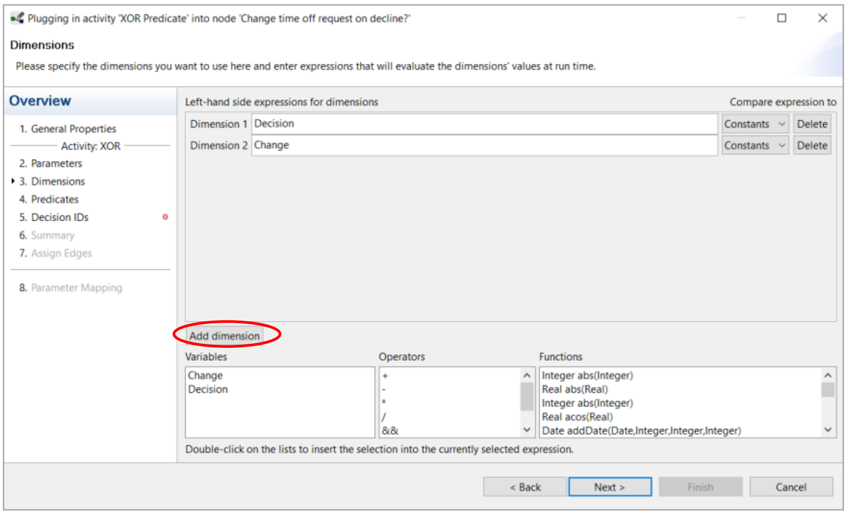

Working on revision 2 (latest)
Main
Identifier
Name
User Form
User Form
de.aristaflow.form.GeneratedForm
de.aristaflow.form.SurveyJS
de.aristaflow.form.WebForm
de.aristaflow.java.StaticJavaCall
de.aristaflow.mail.FormattingMailer
de.aristaflow.mail.SendMail
de.aristaflow.ocr.Tess4J
de.aristaflow.omaccess.OrgModelAc
de.aristaflow.openoffice.BatchIntegr
de.aristaflow.openoffice.Integrator
de.aristaflow.rules.XOR
XOR Predicate
XOR Predicate
de.aristaflow.systemtools.ProcessInt
de.aristaflow.tools.BSH
de.aristaflow.tools.FileIO
de.aristaflow.tools.net.FTP
de.aristaflow.tools.net.HTTP
de.aristaflow.tools.net.SFTP
Begin STRING
Originator STRING
End STRING
Decision BOOLEAN
Change BOOLEAN
Fill out request for time off form
Approve time off request
appro...
declin...
Time off request approved
Update Time off account
Time Off request declined. Change?
0
1

c. Process step “*Time off request approved*”: *de.aristaflow.form.Form* → *User Form* → *User Form*. Enter a description like “*Your request for time off has been approved.*” and finish the wizard.

d. Process step “*Update time off account*”: *de.aristaflow.form.Form* → *User Form* → *User Form*. Enter a description like “Please update the request for time off account for the following person:”, check the mapping (should match already) and finish the wizard.

e. Process step “*Time off request declined. Change?*”: *de.aristaflow.form.Form* → *User Form* → *User Form*. Enter a description like “*Your request for time off was declined. Do you want to change it?*”, check the mapping (should match already) and finish the wizard.

f. Process step “*Change time off request on decline?*”: We have to create a decision on this loop end node, so assign *de.aristaflow.rules.XOR* → *XOR Predicate* → *XOR Predicate*. Create two dimensions one for the “*Change*” and another one for the “*Decision*” input parameter by clicking on Add dimension. Take “Decision” as first dimension and “Change” as second dimension by double click on the parameters.


Plugging in activity 'XOR Predicate' into node 'Change time off request on decline?'

Dimensions

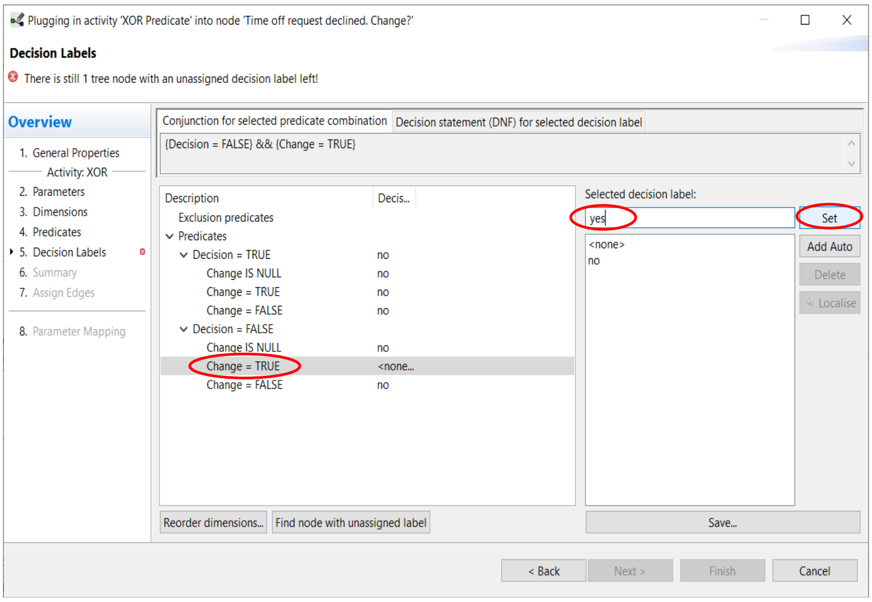
Please specify the dimensions you want to use here and enter expressions that will evaluate the dimensions' values at run time.

Left-hand side expressions for dimensions — Compare expression to

Dimension 1	Decision	Constants	Delete
Dimension 2	Change	Constants	Delete

Double-click on the lists to insert the selection into the currently selected expression.

On the next wizard page click the *Add All* button (bottom right) for the “*Decision*” dimension and switch to the “*Change*” dimension (drop down box at the top) and create the predicates “*Change IS NULL*“, “*Change = TRUE*” and “*Change = FALSE*” for the “*Change*” dimension.


Plugging in activity 'XOR Predicate' into node 'Time off request declined. Change?'

Decision Labels

There is still 1 tree node with an unassigned decision label left!

{Decision = FALSE} && {Change = TRUE}

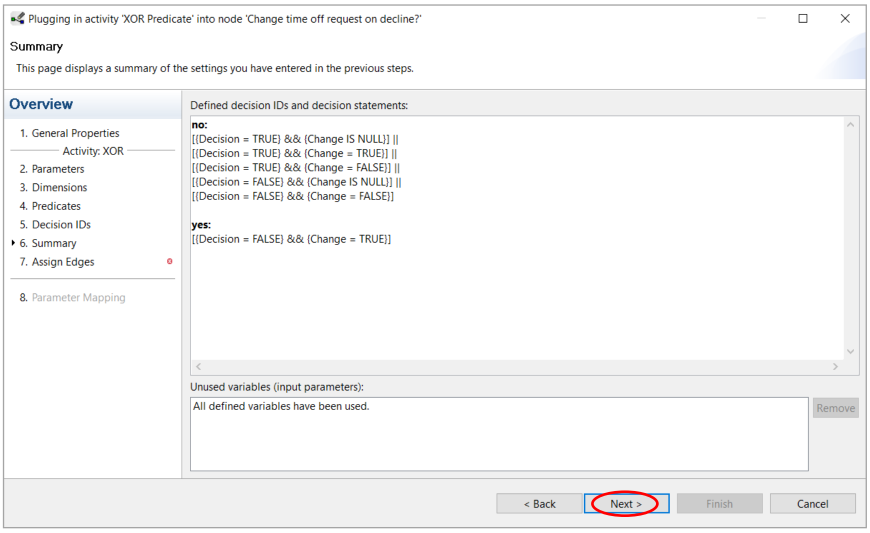
Selected decision label: yes

<none>
no

Description	Decis...
Exclusion predicates	
Predicates	
Decision = TRUE	no
Change IS NULL	no
Change = TRUE	no
Change = FALSE	no
Decision = FALSE	
Change IS NULL	no
Change = TRUE	<none...
Change = FALSE	no

On the *Decision IDs* wizard page make sure the dimensions are ordered in the sequence shown in the screenshot. If not click “*Reorder dimensions*”.

Select “*Decision = TRUE*”, enter “*No*” in the decision ID text box and press the *Set* button. Assign “*No*” to “*Change = FALSE*” and “*Change IS NULL*” and “*Yes*” to “*Change = True*” in the branch under “*Decision = FALSE*”. Check the result on the next page.


Plugging in activity 'XOR Predicate' into node 'Change time off request on decline?'

Summary

This page displays a summary of the settings you have entered in the previous steps.

Defined decision IDs and decision statements:

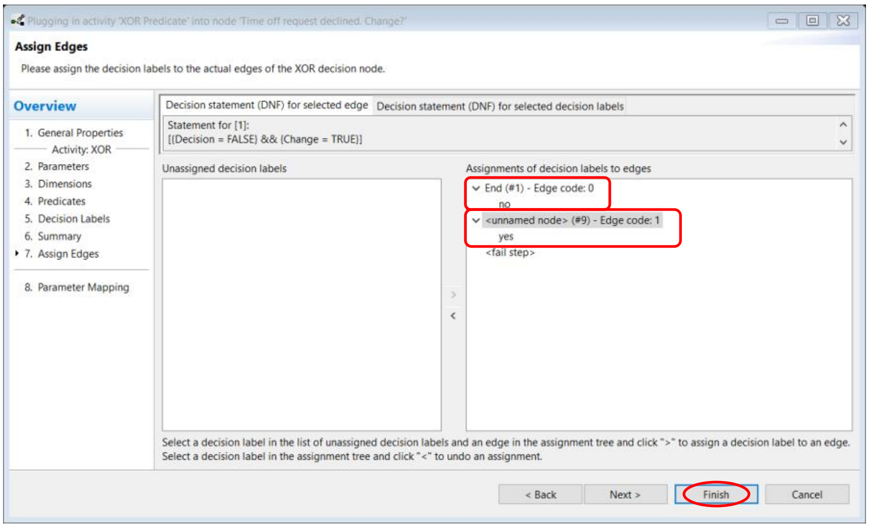
```
no:
[{Decision = TRUE} && {Change IS NULL}] ||
[{Decision = TRUE} && {Change = TRUE}] ||
[{Decision = TRUE} && {Change = FALSE}] ||
[{Decision = FALSE} && {Change IS NULL}] ||
[{Decision = FALSE} && {Change = FALSE}]

yes:
[{Decision = FALSE} && {Change = TRUE}]
```

Unused variables (input parameters):

All defined variables have been used.

Assign “*No*” to the branch with the End node to leave the loop and “*Yes*” to jump back to the beginning of the loop. Finish the wizard; the parameter mapping should be correct already.



Plugging in activity 'XOR Predicate' into node 'Time off request declined. Change?'

Assign Edges

Please assign the decision labels to the actual edges of the XOR decision node.

Decision statement (DNF) for selected edge

Statement for [1]:
[(Decision = FALSE) && (Change = TRUE)]

Assignments of decision labels to edges

- End (#1) - Edge code: 0
  - no
- <unnamed node> (#9) - Edge code: 1
  - yes
- <fail step>

Select a decision label in the list of unassigned decision labels and an edge in the assignment tree and click ">" to assign a decision label to an edge.
Select a decision label in the assignment tree and click "<" to undo an assignment.

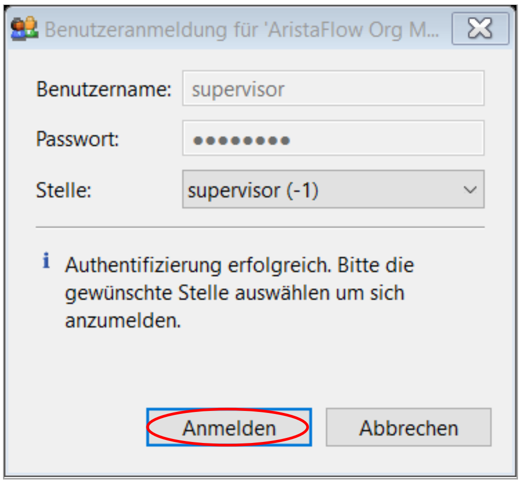
With the assignment of the decision activity all relevant process steps are provided with functionality. The last step to complete our "*Request for time off*" process is the staff assignment for the single process steps.

Assign staff to the process steps

In this chapter you learn how to manage your organizational model with the AristaFlow *Org Model Manager* and how to use this model to assign staff or more precisely staff assignment rules to the process steps. This is necessary for a correct routing of the process steps' work items to user worklists at runtime.

We begin with the creation of the org model entities necessary for our "*Request for time off*" process. Note that the org model is completely separated from any kind of process, i.e. entities which have been created in the org model can be used in arbitrary business processes.

Start the AristaFlow *Org. Model Editor* from the Windows start menu under ... → AristaFlow *BPM Suite* → *Org. Model Editor* or in your installation folder *(…\OrgModelEditor\OrgModelEditor.exe)* and login with supervisor / password.



Benutzername: supervisor

Passwort: ●●●●●●●●

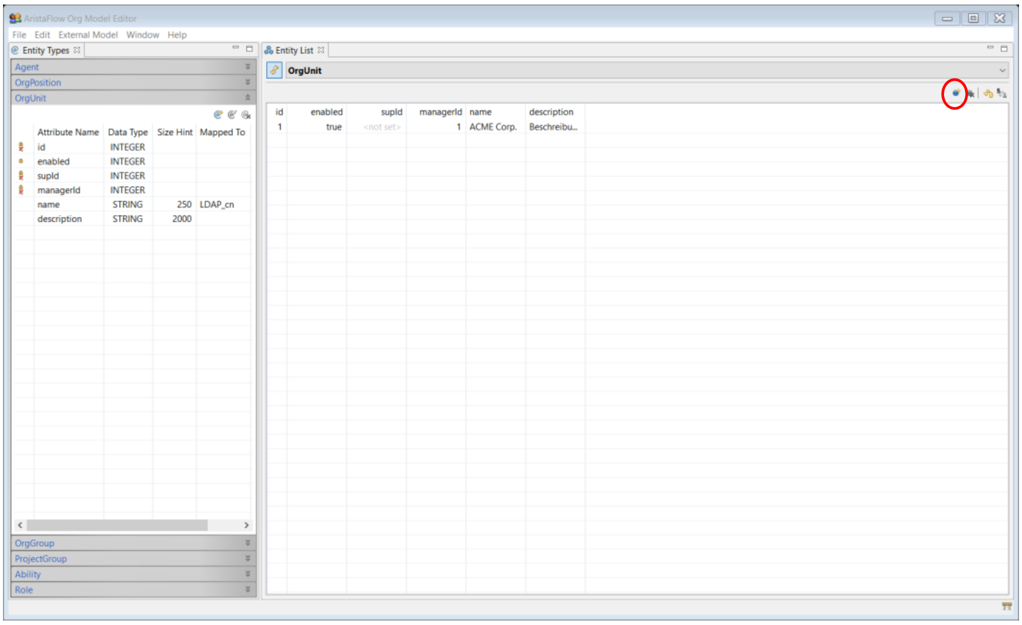
Stelle: supervisor (-1)

Authentifizierung erfolgreich. Bitte die gewünschte Stelle auswählen um sich anzumelden.

For the "*Request for time off*" process we create the following organizational entities as an example:

a. An organizational unit (*OrgUnit*) where the originator belongs to (e.g. "*purchase*").

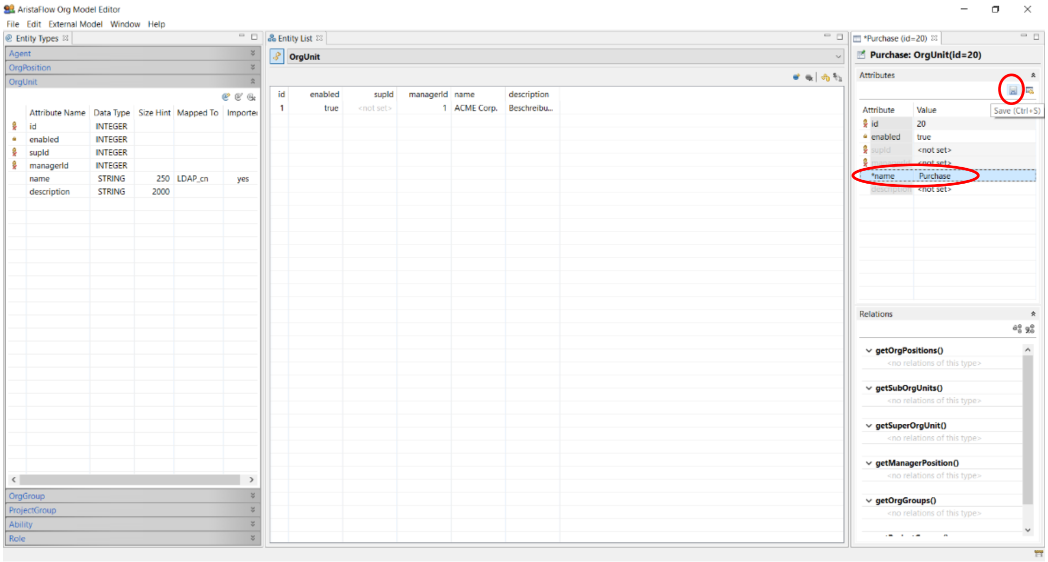
   Select *OrgUnit* in the Entity Types view on the left hand side of the editor and push the *Add Entity* button (see screenshot).



Attribute Name	Data Type	Size Hint	Mapped To
id	INTEGER		
enabled	INTEGER		
supId	INTEGER		
managerId	INTEGER		
name	STRING	250	LDAP_cn
description	STRING	2000	

id	enabled	supId	managerId	name	description
1	true	<not set>	1	ACME Corp.	Beschreibu...

The *new OrgUnit* view opens on the right hand side. Here you can enter or change the attributes of an entity. In our case enter “*Purchase*” in the input field next to name and click the save button with the disk symbol.

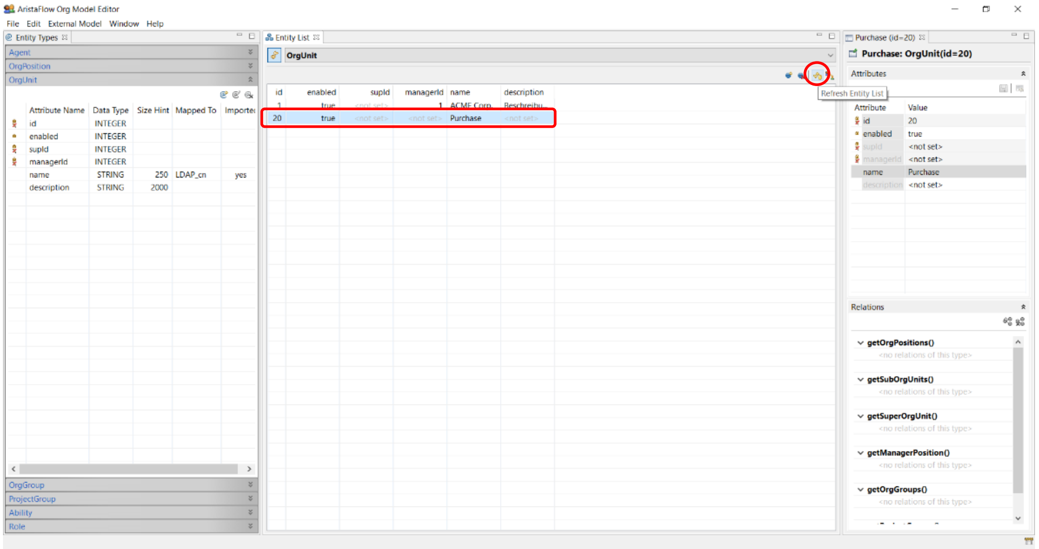


Attribute Name	Data Type	Size Hint	Mapped To	Importe
id	INTEGER			
enabled	INTEGER			
supId	INTEGER			
managerId	INTEGER			
name	STRING	250	LDAP_cn	yes
description	STRING	2000		

id	enabled	supId	managerId	name	description
1	true	<not set>	1	ACME Corp.	Beschreibu...

Attribute	Value
id	20
enabled	true
supId	<not set>
managerId	<not set>
*name	Purchase
description	<not set>

Update the *Entity List* view in the center of the *OrgModelEditor* with the *Refresh* button (see screenshot).

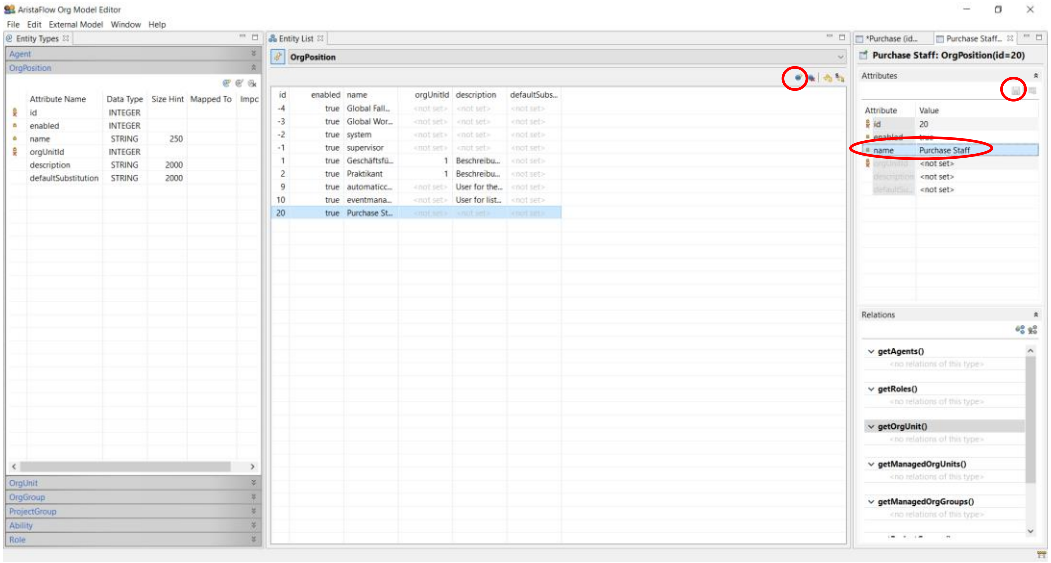


Attribute Name	Data Type	Size Hint	Mapped To	Importe
id	INTEGER			
enabled	INTEGER			
supId	INTEGER			
managerId	INTEGER			
name	STRING	250	LDAP_cn	yes
description	STRING	2000		

id	enabled	supId	managerId	name	description
1	true	<not set>	1	ACME Corp.	Beschreibu...
20	true	<not set>	<not set>	Purchase	<not set>

Attribute	Value
id	20
enabled	true
supId	<not set>
managerId	<not set>
name	Purchase
description	<not set>

b. Create the *OrgPosition* “*Purchase Staff*” and assign it to the OrgUnit “*Purchase*”. More precisely select the *OrgPosition* in the *Entity Types* view. Create a new entity, enter the attribute name “*Purchase Staff*” and save it with the disk symbol.



Attribute Name	Data Type	Size Hint	Mapped To	Impc
id	INTEGER			
enabled	INTEGER			
name	STRING	250		
orgUnitId	INTEGER			
description	STRING	2000		
defaultSubstitution	STRING	2000		

OrgPosition

id	enabled	name	orgUnitId	description	defaultSubs...
-4	true	Global Fall...	<not set>	<not set>	<not set>
-3	true	Global Wor...	<not set>	<not set>	<not set>
-2	true	system	<not set>	<not set>	<not set>
-1	true	supervisor	<not set>	<not set>	<not set>
1	true	Geschäftsfü...	1	Beschreibu...	<not set>
2	true	Praktikant	1	Beschreibu...	<not set>
9	true	automaticc...	<not set>	User for the...	<not set>
10	true	eventmana...	<not set>	User for list...	<not set>
20	true	Purchase St...	<not set>	<not set>	<not set>

Purchase Staff: OrgPosition(id=20)

Attribute	Value
id	20
enabled	true
name	Purchase Staff
orgUnitId	<not set>
description	<not set>
defaultSu...	<not set>

getAgents()
<no relations of this type>

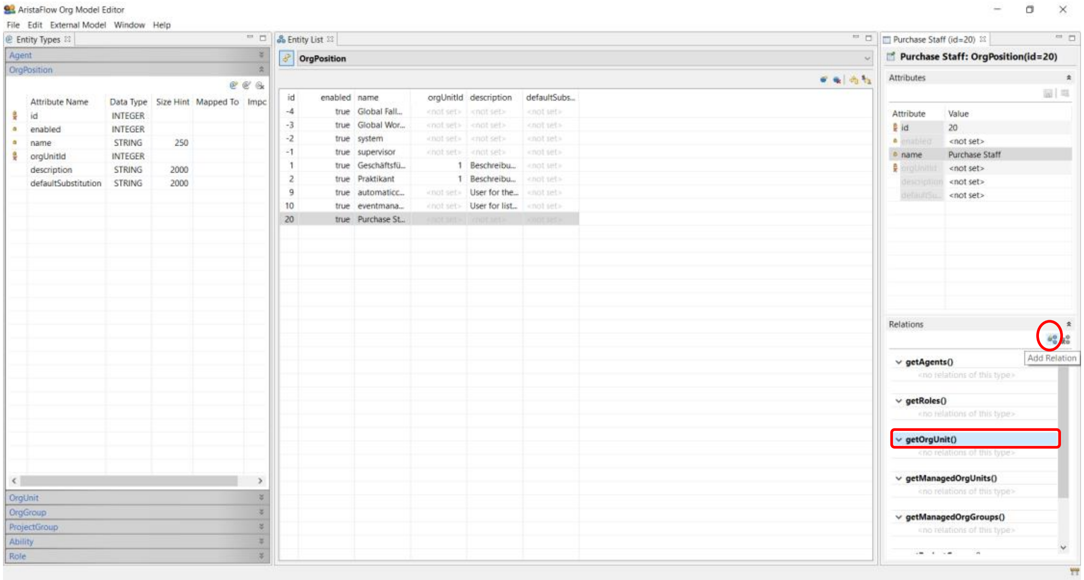
getRoles()
<no relations of this type>

getOrgUnit()
<no relations of this type>

getManagedOrgUnits()
<no relations of this type>

getManagedOrgGroups()
<no relations of this type>

Now we have to connect the *OrgPosition* “*Purchase Staff*” with the *OrgUnit* “*Purchase*”. Switch to the *Relations* view on the right hand side, select *getOrgUnit()* and click the *Add Relation* button (see screenshot).



Attribute Name	Data Type	Size Hint	Mapped To	Impc
id	INTEGER			
enabled	INTEGER			
name	STRING	250		
orgUnitId	INTEGER			
description	STRING	2000		
defaultSubstitution	STRING	2000		

OrgPosition

id	enabled	name	orgUnitId	description	defaultSubs...
-4	true	Global Fall...	<not set>	<not set>	<not set>
-3	true	Global Wor...	<not set>	<not set>	<not set>
-2	true	system	<not set>	<not set>	<not set>
-1	true	supervisor	<not set>	<not set>	<not set>
1	true	Geschäftsfü...	1	Beschreibu...	<not set>
2	true	Praktikant	1	Beschreibu...	<not set>
9	true	automaticc...	<not set>	User for the...	<not set>
10	true	eventmana...	<not set>	User for list...	<not set>
20	true	Purchase St...	<not set>	<not set>	<not set>

Purchase Staff: OrgPosition(id=20)

Attribute	Value
id	20
enabled	<not set>
name	Purchase Staff
orgUnitId	<not set>
description	<not set>
defaultSu...	<not set>

getAgents()
<no relations of this type>

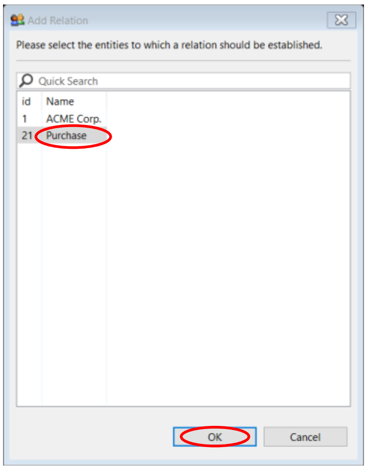
getRoles()
<no relations of this type>

getOrgUnit()
<no relations of this type>

getManagedOrgUnits()
<no relations of this type>

getManagedOrgGroups()
<no relations of this type>

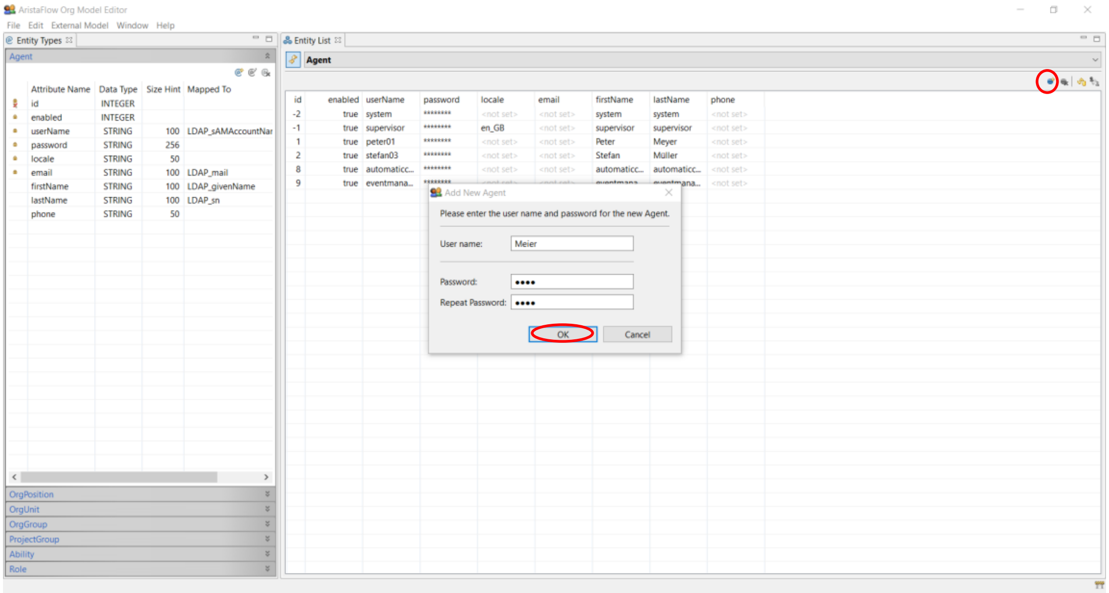
Choose “*Purchase*“ in the upcoming dialogue and finish with the *OK* button.



Please select the entities to which a relation should be established.

id	Name
1	ACME Corp.
21	Purchase

c. Create the persons (*Agents*) and assign it to the *OrgPosition “Purchase Staff”*. To do this select *Agent* in the *Entity Types* and press the *New Entity* button. Enter a *User name* and a *Password* in the upcoming dialogue (e. g. Meier / test).



Attribute Name	Data Type	Size Hint	Mapped To
id	INTEGER		
enabled	INTEGER		
userName	STRING	100	LDAP_sAMAccountNar
password	STRING	256	
locale	STRING	50	
email	STRING	100	LDAP_mail
firstName	STRING	100	LDAP_givenName
lastName	STRING	100	LDAP_sn
phone	STRING	50	

id	enabled	userName	password	locale	email	firstName	lastName	phone
-2	true	system	********	<not set>	<not set>	system	system	<not set>
-1	true	supervisor	********	en_GB	<not set>	supervisor	supervisor	<not set>
1	true	peter01	********	<not set>	<not set>	Peter	Meyer	<not set>
2	true	stefan03	********	<not set>	<not set>	Stefan	Müller	<not set>
8	true	automaticc...	********	<not set>	<not set>	automaticc...	automaticc...	<not set>
9	true	eventmana...	********	<not set>	<not set>	eventmana...	eventmana...	<not set>

Add New Agent

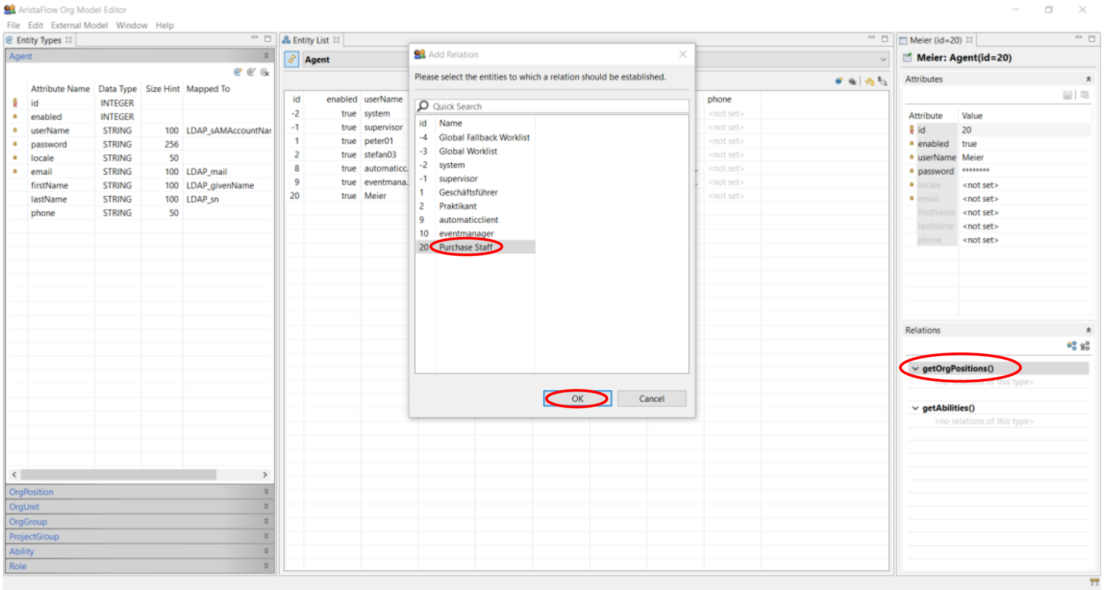
Please enter the user name and password for the new Agent.

User name: Meier

Password: ••••

Repeat Password: ••••

Now we have to connect the *Agent “Meier”* with the *OrgPosition “Purchase Staff”*. Switch to the *Relations* view on the right hand side, select *getOrgPositions()* and click the *Add Relation* button (see screenshot) or double click on *getOrgPositions()*.



id	enabled	userName	phone
-2	true	system	<not set>
-1	true	supervisor	<not set>
1	true	peter01	<not set>
2	true	stefan03	<not set>
8	true	automaticc...	<not set>
9	true	eventmana...	<not set>
20	true	Meier	<not set>

Add Relation

Please select the entities to which a relation should be established.

Quick Search

id	Name
-4	Global Fallback Worklist
-3	Global Worklist
-2	system
-1	supervisor
1	Geschäftsführer
2	Praktikant
9	automaticclient
10	eventmanager
20	Purchase Staff

Meier: Agent(id=20)

Attributes

Attribute	Value
id	20
enabled	true
userName	Meier
password	********
locale	<not set>
email	<not set>
firstName	<not set>
lastName	<not set>
phone	<not set>

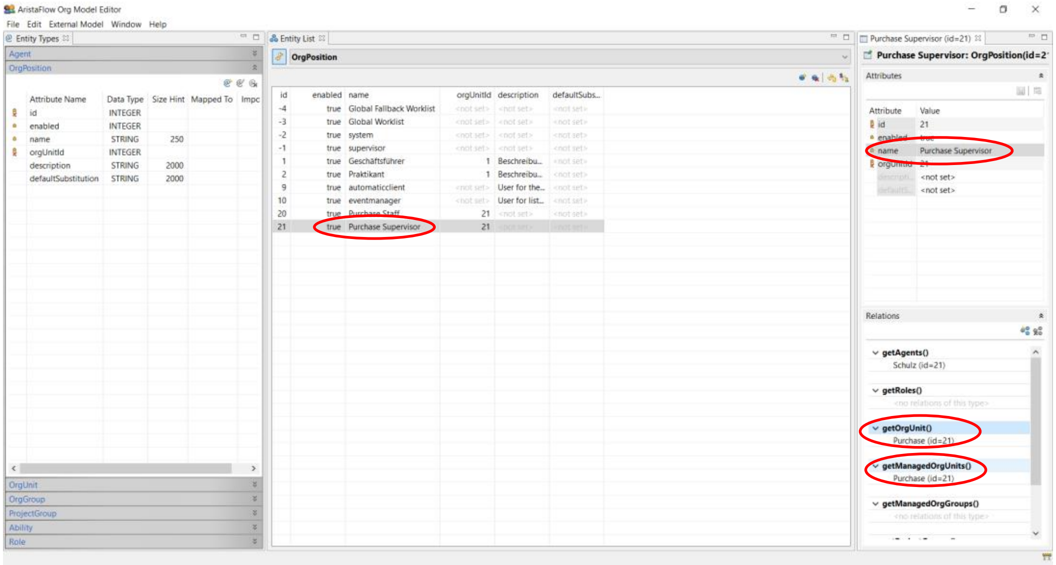
Relations

getOrgPositions()
<no relations of this type>

getAbilities()
<no relations of this type>

Choose “*Purchase Staff*“ in the upcoming dialogue and finish with the OK button.

d. Now we want to create the person which is the manager of the “*Purchase*” *OrgUnit* and because of that the supervisor of all “*Purchase Staff*”. Therefore create the *OrgPosition* “*Purchase Supervisor*” and assign it to the “*Purchase*” *OrgUnit* by double-clicking *getOrgUnit()* in the *“Relations”* view and selecting “*Purchase”* in the upcoming dialogue. Then create a second relation between *“Purchase”* and *“Purchase Supervisor”* and link them through the manager position. Double-click *getManagedOrgUnits()* and choose *“Purchase”*.



AristaFlow Org Model Editor
File Edit External Model Window Help
Entity Types
Agent
OrgPosition
OrgUnit
OrgGroup
ProjectGroup
Ability
Role
Entity List
Purchase Supervisor (id=21)

Attribute Name	Data Type	Size Hint	Mapped To	Impc
id	INTEGER			
enabled	INTEGER			
name	STRING	250		
orgUnitId	INTEGER			
description	STRING	2000		
defaultSubstitution	STRING	2000		

OrgPosition

id	enabled	name	orgUnitId	description	defaultSubs...
-4	true	Global Fallback Worklist	<not set>	<not set>	<not set>
-3	true	Global Worklist	<not set>	<not set>	<not set>
-2	true	system	<not set>	<not set>	<not set>
-1	true	supervisor	<not set>	<not set>	<not set>
1	true	Geschäftsführer	1	Beschreibu...	<not set>
2	true	Praktikant	1	Beschreibu...	<not set>
9	true	automaticclient	<not set>	User for the...	<not set>
10	true	eventmanager	<not set>	User for list...	<not set>
20	true	Purchase Staff	21	<not set>	<not set>
21	true	Purchase Supervisor	21	<not set>	<not set>

Purchase Supervisor: OrgPosition(id=2

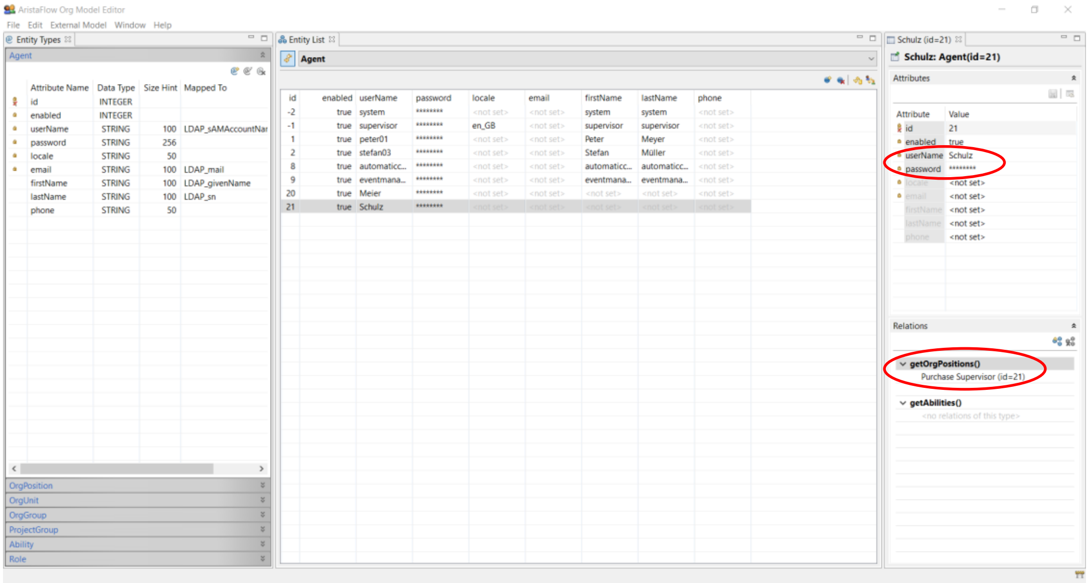
Attributes

Attribute	Value
id	21
enabled	true
name	Purchase Supervisor
orgUnitId	21
descripti...	<not set>
defaultS...	<not set>

Relations

- getAgents()
  - Schulz (id=21)
- getRoles()
  - <no relations of this type>
- getOrgUnit()
  - Purchase (id=21)
- getManagedOrgUnits()
  - Purchase (id=21)
- getManagedOrgGroups()
  - <no relations of this type>

Finally create the *Agent* (e. g. Schulz / test) and assign it to the "*Purchase Supervisor*" by *getOrgPositions()*.



Attribute Name	Data Type	Size Hint	Mapped To
id	INTEGER		
enabled	INTEGER		
userName	STRING	100	LDAP_sAMAccountNar
password	STRING	256	
locale	STRING	50	
email	STRING	100	LDAP_mail
firstName	STRING	100	LDAP_givenName
lastName	STRING	100	LDAP_sn
phone	STRING	50	

Agent

id	enabled	userName	password	locale	email	firstName	lastName	phone
-2	true	system	********	<not set>	<not set>	system	system	<not set>
-1	true	supervisor	********	en_GB	<not set>	supervisor	supervisor	<not set>
1	true	peter01	********	<not set>	<not set>	Peter	Meyer	<not set>
2	true	stefan03	********	<not set>	<not set>	Stefan	Müller	<not set>
8	true	automaticc...	********	<not set>	<not set>	automaticc...	automaticc...	<not set>
9	true	eventmana...	********	<not set>	<not set>	eventmana...	eventmana...	<not set>
20	true	Meier	********	<not set>	<not set>	<not set>	<not set>	<not set>
21	true	Schulz	********	<not set>	<not set>	<not set>	<not set>	<not set>

Schulz: Agent(id=21)

Attributes

Attribute	Value
id	21
enabled	true
userName	Schulz
password	********
locale	<not set>
email	<not set>
firstName	<not set>
lastName	<not set>
phone	<not set>

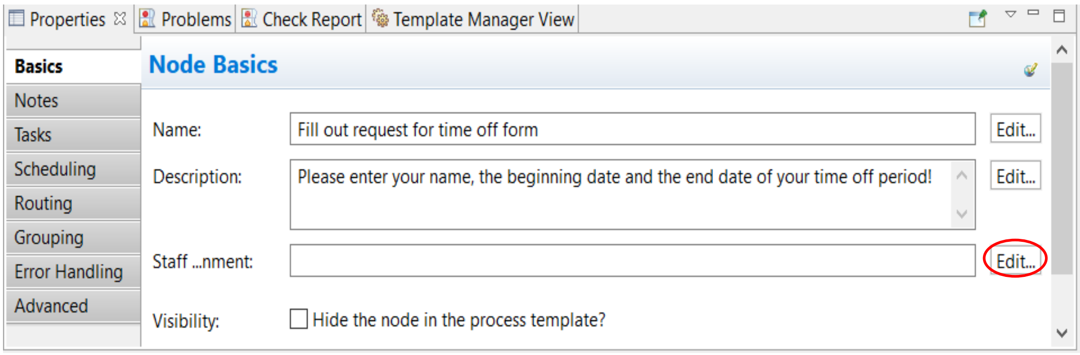
Relations

- getOrgPositions()
  - Purchase Supervisor (id=21)
- getAbilities()
  - <no relations of this type>

e. To complete our *OrgModel* for our example, we have to create a person which is responsible for the update of the time off account including related *OrgPositions* and *OrgUnits*. Create an *OrgUnit "Human Resource"*, an *OrgPosition "Human Resource Staff"* and an *Agent* (e.g. Schmidt / test). Create the relations between the entities in accordance to the descriptions given in the former sections (*"Human Resource Staff" -> getOrgUnit() = "Human Resource"* and *Schmidt -> getOrgPositions() = "Human Resource Staff"*).

Since we are finished with the creation of all the *OrgUnits*, *OrgPositions* and *Agents* which are sufficient for our *"Request for time off"* process we are now ready to use this entities to assign the staff to our process steps. Switch back to the *AristaFlow Process Template Editor*.

a. Select the process step where you want to assign staff. Click the *Edit…* button next to the Staff Assignment input field in the *Basics* tab of the *Properties View*.



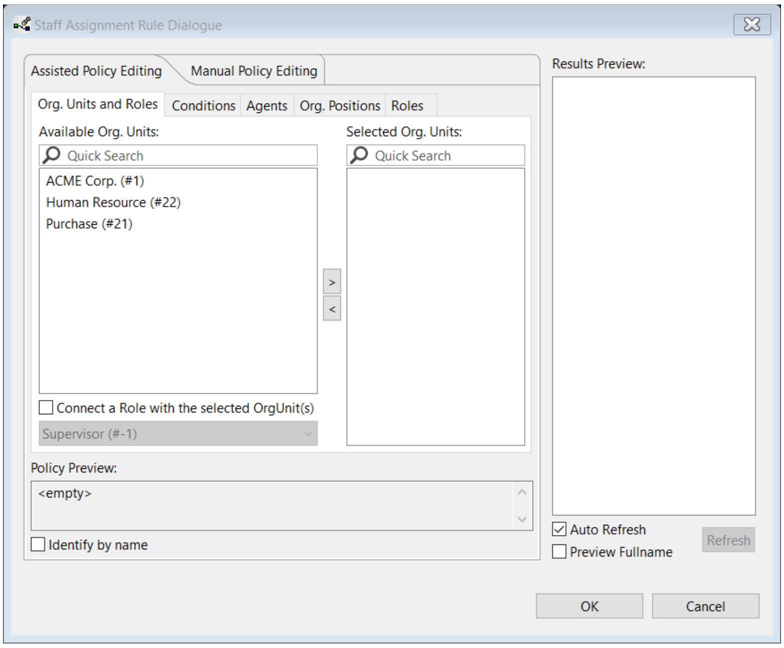
Name: Fill out request for time off form

Description: Please enter your name, the beginning date and the end date of your time off period!

Staff ...nment:

Visibility: □ Hide the node in the process template?

The *Staff Assignment Rule* Wizard opens.



Available Org. Units:
ACME Corp. (#1)
Human Resource (#22)
Purchase (#21)

□ Connect a Role with the selected OrgUnit(s)

Supervisor (#-1)

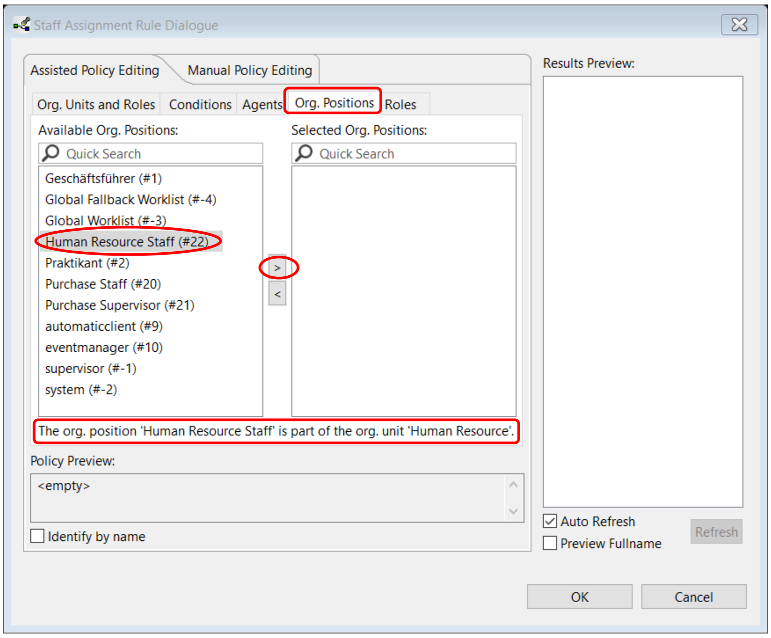
Policy Preview:
<empty>

□ Identify by name

☑ Auto Refresh

□ Preview Fullname

As an example we assign the staff or more precisely the staff assignment rule for the process step "*Update time off account*". Switch to the *OrgPositions* tab and select "*Human Resource Staff*". The *OrgUnit* where the *OrgPosition* belongs to is shown automatically.



Available Org. Positions:
Geschäftsführer (#1)
Global Fallback Worklist (#-4)
Global Worklist (#-3)
Human Resource Staff (#22)
Praktikant (#2)
Purchase Staff (#20)
Purchase Supervisor (#21)
automaticclient (#9)
eventmanager (#10)
supervisor (#-1)
system (#-2)

The org. position 'Human Resource Staff' is part of the org. unit 'Human Resource'.

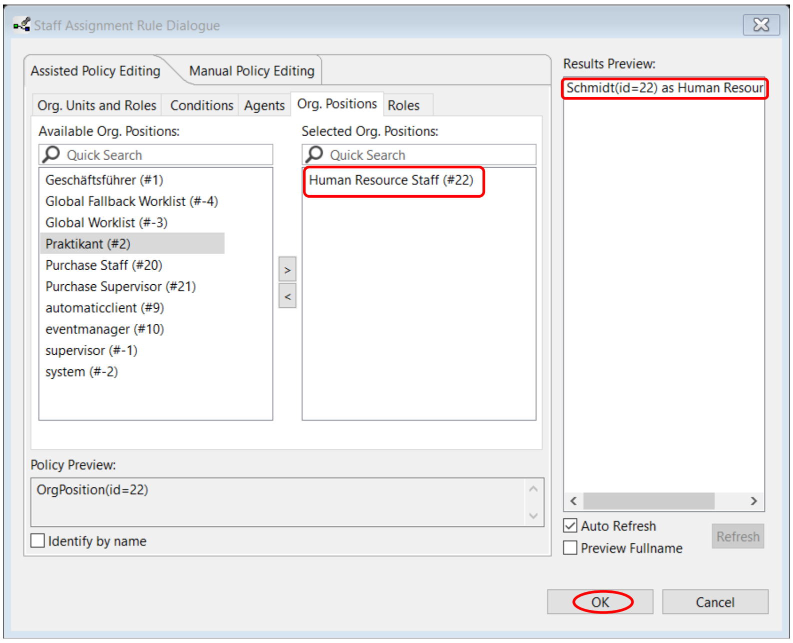
Policy Preview:
<empty>

□ Identify by name

☑ Auto Refresh

□ Preview Fullname

Push the „>“ button. The “*Human Resources Staff”* has been moved from the *Available Org. Positions* list to the *Selected Org. Positions* list. In the *Results Preview* you can see all persons which are members of the selected org model entities (in this case the *OrgPosition “Human Resources Staff”*).



Available Org. Positions:

Geschäftsführer (#1)
Global Fallback Worklist (#-4)
Global Worklist (#-3)
Praktikant (#2)
Purchase Staff (#20)
Purchase Supervisor (#21)
automaticclient (#9)
eventmanager (#10)
supervisor (#-1)
system (#-2)

Selected Org. Positions:

Human Resource Staff (#22)

Results Preview:

Schmidt(id=22) as Human Resour

Policy Preview:

OrgPosition(id=22)

☐ Identify by name

☑ Auto Refresh

☐ Preview Fullname

For the “*Update time off account*” step this is sufficient so close the wizard with the *OK* button.

b. Assign the staff according to the explanation in a. to the following process steps:

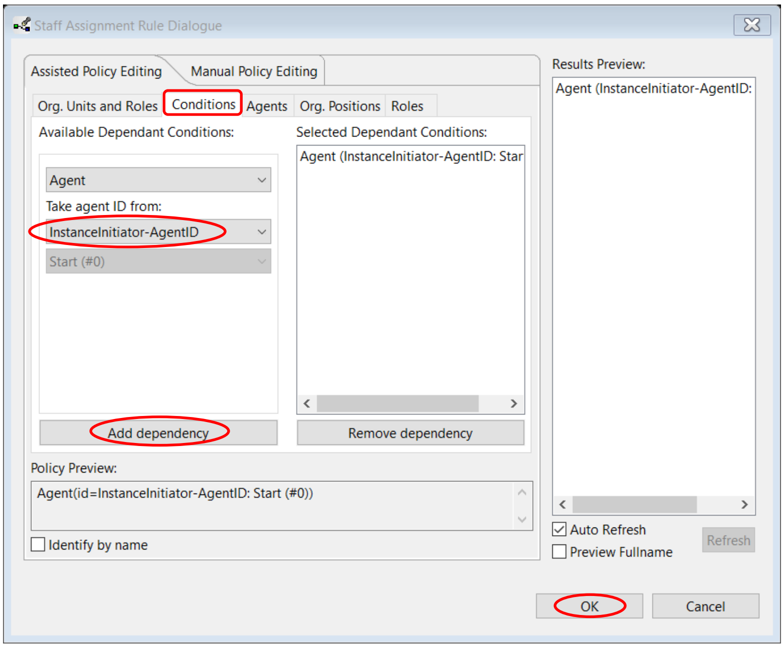
  i. Process step “*approved / declined*”: Switch to tab *OrgPositions* and select “*automaticclient*”.
  The “*automaticclient*” is a client which acts similar to a normal user client with the difference that the “*automaticclient*” executes services / activities assigned to (GUI less) process steps on server side immediately after the steps have been assigned to its worklist. This is a special feature of the AristaFlow *BPM-Suite* and offers possibilities like simple load balancing by installing automatic clients on different servers.

  ii. Process step “*Time off request declined. Change?*”: Switch to tab *OrgPositions* and select “*automaticclient*”.

  iii. The process steps which are used to maintain the process structure like the loop start node (next to the process Start node) and the corresponding join node of the “*approved / declined*” XOR split do not need a *staff assignment rule*. The server executes them automatically.

c. Apart from the assignment of simple org model entities like *OrgPositions*, *Agents*, *OrgUnits*, *Roles*, etc. the *Staff Assignment Rule Wizard* offers the possibility to model complex dependencies between the performers of different process steps. For example:

  i. Process step “*Fill out request for time off form*”: Switch to the tab *Conditions* select *Agent* in the upper drop down box and choose *InstanceInitiator-AgentID* in the *Take agent ID from*: drop down box. Push the *Add dependency* button. Now the created staff assignment rule is configured to choose exactly the person as an executor for this process step at runtime which has started the whole process instance.



Staff Assignment Rule Dialogue

Assisted Policy Editing | Manual Policy Editing

Org. Units and Roles | Conditions | Agents | Org. Positions | Roles

Available Dependant Conditions:

Agent

Take agent ID from:

InstanceInitiator-AgentID

Start (#0)

Selected Dependant Conditions:

Agent (InstanceInitiator-AgentID: Star

Add dependency | Remove dependency

Policy Preview:

Agent(id=InstanceInitiator-AgentID: Start (#0))

☐ Identify by name

Results Preview:

Agent (InstanceInitiator-AgentID:

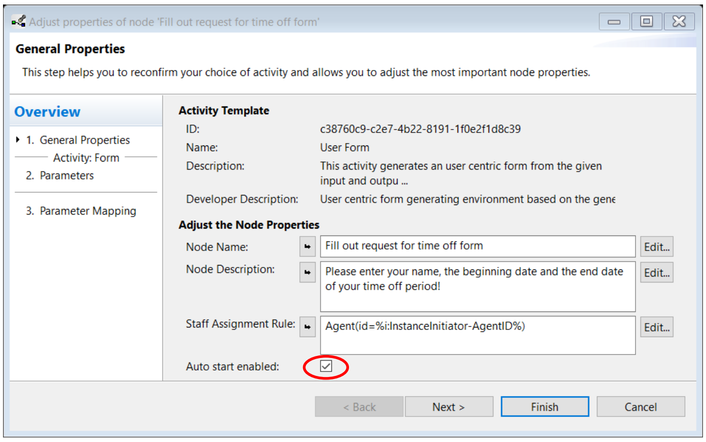
☑ Auto Refresh

☐ Preview Fullname

Refresh

OK | Cancel

Hint:
Since the performer of the process step “*Fill out request for time off form*” is the starter of the process instance at runtime we can link the execution of this step to the start event of the process instance. As a result the step “*Fill out request for time off form*” is executed automatically immediately after process start at runtime.
To do so double-click the “*Fill out request for time off form*” step and check the



Adjust properties of node 'Fill out request for time off form'

General Properties

This step helps you to reconfirm your choice of activity and allows you to adjust the most important node properties.

Overview
- 1. General Properties
  Activity: Form
- 2. Parameters
- 3. Parameter Mapping

Activity Template

ID: c38760c9-c2e7-4b22-8191-1f0e2f1d8c39
Name: User Form
Description: This activity generates an user centric form from the given input and outpu ...
Developer Description: User centric form generating environment based on the gene

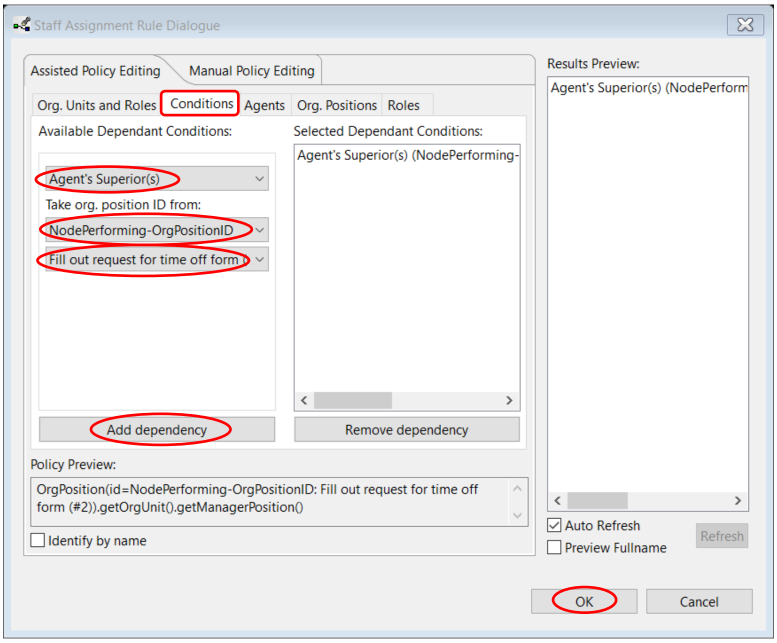
Adjust the Node Properties

Node Name: Fill out request for time off form
Node Description: Please enter your name, the beginning date and the end date of your time off period!
Staff Assignment Rule: Agent(id=%i:InstanceInitiator-AgentID%)
Auto start enabled: ☑

< Back | Next > | Finish | Cancel

ii. Assign the staff assignment rules for the process steps *“Time off request approved“* and *“Time off request declined. Change?”* in accordance to i. but without the auto start option.

iii. Process step “*Approve time off request*”: Switch to tab *Conditions*, select *Agent’s Superior(s)* in the first, *NodePerforming-OrgPositionID* in the second and *Fill out request for time off form (#2)* in the third drop down box. Push the add dependency button. Now the staff assignment rule is configured to always delegate the process step *“Approve time off request”* exactly to the worklist of the “*Fill out request for time off form”* step performer’s supervisor.



Available Dependant Conditions:

Agent's Superior(s)

Take org. position ID from:

NodePerforming-OrgPositionID

Fill out request for time off form

Selected Dependant Conditions:

Agent's Superior(s) (NodePerforming-

Policy Preview:

OrgPosition(id=NodePerforming-OrgPositionID: Fill out request for time off form (#2)).getOrgUnit().getManagerPosition()

□ Identify by name

Results Preview:

Agent's Superior(s) (NodePerform

☑ Auto Refresh

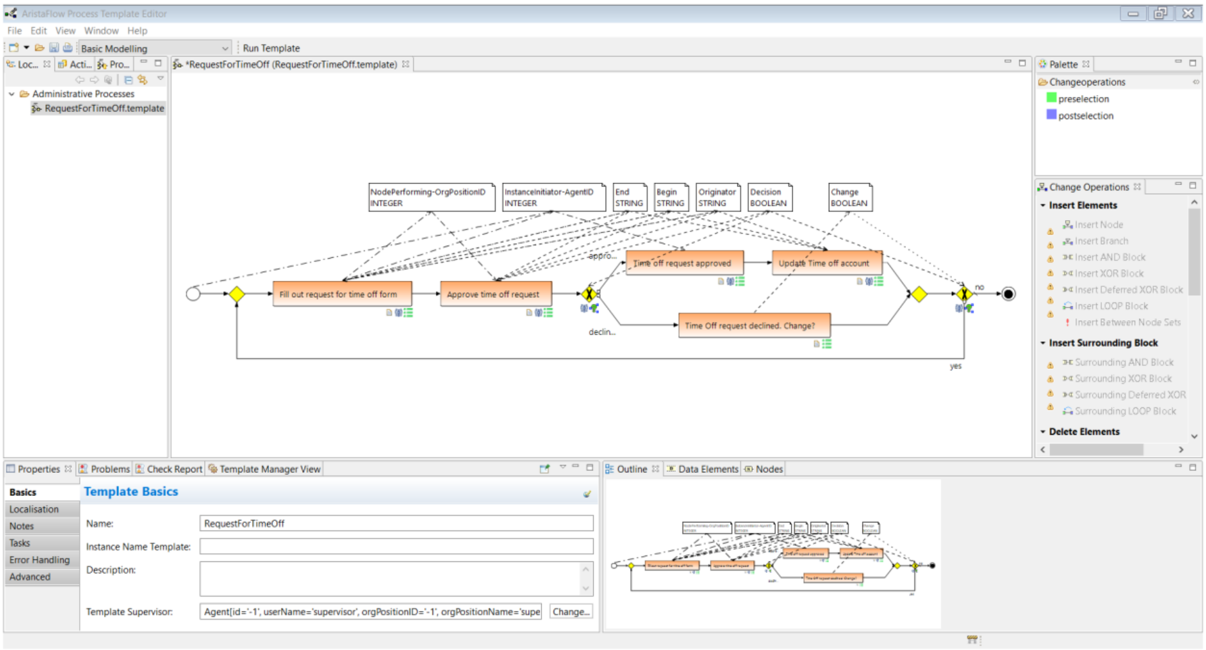
□ Preview Fullname

Hint:
Right-click the working area of the editor and select *Display options*. Check *System Data Elements Visible* and *System Data Edges Visible* in the upcoming dialogue to visualize the dependant staff assignment rules.

Hint:
The *Staff Assignment Rule Wizard* offers the possibility to create arbitrary complex staff assignment rules in the *Manual Policy Editing* tab. In this mode you have full access to all entities and functions of the *AristaFlow OrgModel* (e.g. project groups, abilities, etc.).

With the assignment of the staff assignment rules to the process steps we have finished the design of our *"Request for time off"* process.

The final result of the whole model is shown in the following screenshot:



Name: RequestForTimeOff

Instance Name Template:

Description:

Template Supervisor: Agent[id='-1', userName='supervisor', orgPositionID='-1', orgPositionName='supe

Hint:
For experienced users the modeling of business processes can be accelerated significantly by dropping the activities from the *Activity Repository* directly on edges instead of already existent process steps. All important properties can be defined in the upcoming wizard and a new process step is inserted at the position of the edge automatically when you close it.

We would be able to deploy our *"Request for time off"* process already, but we want to make sure that the semantics of the modeled process it correct. Therefore we introduce the *AristaFlow Test Client* in the next section.

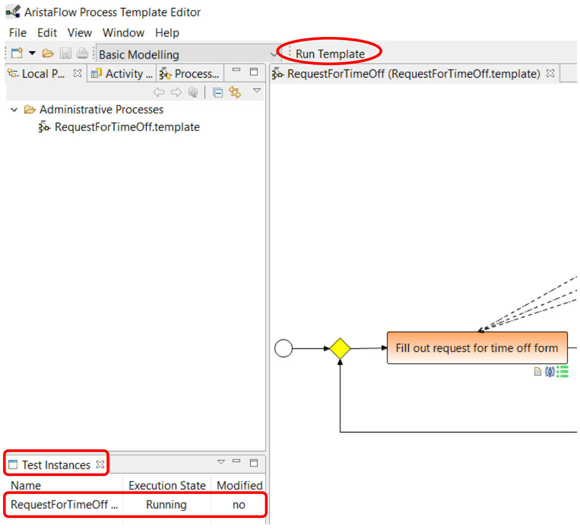
Validate the semantics of the process

As you have learned in the former sections the syntactical correctness of a process is checked automatically by the AristaFlow system during modeling. To check its semantical correctness, what means to check if the modeled process does really what it is supposed to do, we can use the *AristaFlow Test Client* to directly execute our business process on an embedded, non-persistent instance of the *AristaFlow Server*.

Note that you do not need to fully model your business process to execute it in the test client. You can do it in any state, no matter if you have created the data flow or if you have assigned services to the steps or not. So it can be perfectly used to show a customer how the process behaves at runtime at a very early stage.

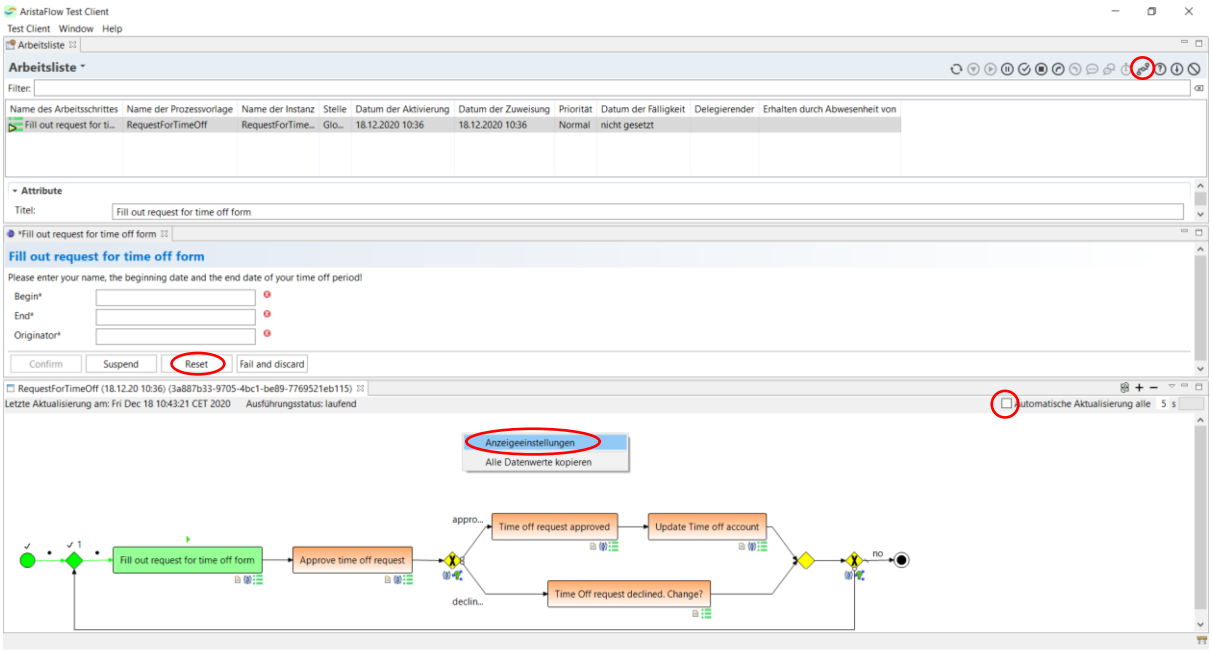
To execute the *"RequestForTimeOff"* process on the *Test Client* be sure that the "RequestForTimeOff" template is opened in the *Process Template Editor* and click the *Run Template* button at the top of the editor to open the process in the test client. A test instance is created automatically which is displayed in the Process Template Editor under *"Test Instances"* (on the left hand center). Now you can track the process behavior using this instance in the test client.

Alternatively you can start the *AristaFlow Test Client* from the Windows start menu under ... → *AristaFlow BPM Suite* → *Test Client* or in your installation folder (*...\TestClient\TestClient.exe*), but this is little more cumbersome since you have to load the template which you want to test by hand.


AristaFlow Process Template Editor
File Edit View Window Help
Basic Modelling
Run Template
Local P...
Activity ...
Process...
RequestForTimeOff (RequestForTimeOff.template)
Administrative Processes
RequestForTimeOff.template
Fill out request for time off form
Test Instances

Name	Execution State	Modified
RequestForTimeOff ...	Running	no

The process is started automatically and you can see that the process step *"Fill out Request for Time off Form"* is activated in the worklist. The process steps *Start* and our loop step are only structuring steps which do not have an activity assigned so they are executed by the process engine. To open the real time monitoring of the process press the *process chains icon* in the upper right corner. You can also enable the automatic update for the execution status by ticking the box on the right above the graph (see screenshot).



Name des Arbeitsschrittes	Name der Prozessvorlage	Name der Instanz	Stelle	Datum der Aktivierung	Datum der Zuweisung	Priorität	Datum der Fälligkeit	Delegierender	Erhalten durch Abwesenheit von
Fill out request for ti...	RequestForTimeOff	RequestForTime...	Glo...	18.12.2020 10:36	18.12.2020 10:36	Normal	nicht gesetzt		

Titel: Fill out request for time off form

Fill out request for time off form

Please enter your name, the beginning date and the end date of your time off period!

Begin*

End*

Originator*

RequestForTimeOff (18.12.20 10:36) (3a887b33-9705-4bc1-be89-7769521eb115)

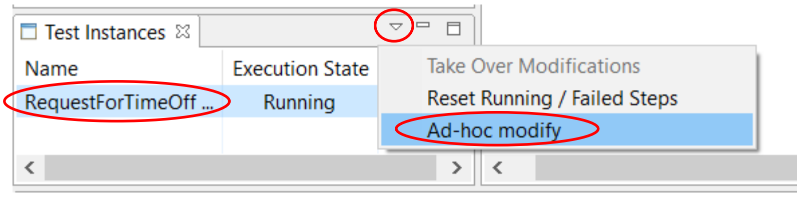
Letzte Aktualisierung am: Fri Dec 18 10:43:21 CET 2020 Ausführungsstatus: laufend

□ Automatische Aktualisierung alle 5 s

Hint:
To display the execution status as usual with BPMN symbols, select *BPMN Symbols* from the dropdown box in the right-click menu over the graph (see the last screenshot).

If you have auto start enabled for the process step *"Fill out request for time off form"* during staff assignment, then the activity which we have assigned to this process step (*User form*) is started automatically. Otherwise to execute the first process step simply double-click the worklist entry.

As we can see, the string input fields in the given form are not very well suited to fill in a *Begin* and *End* date. So we close the form with the *reset* button and we switch back to the *Process Template Editor*.
Select the *"RequestForTimeOff"* process in the *Test Instances* view and chose *Ad-hoc modify* in the menu.

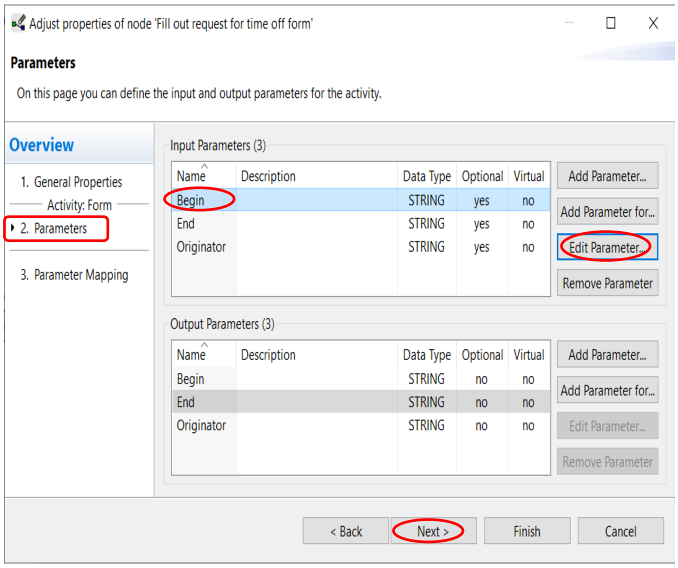


Name	Execution State
RequestForTimeOff ...	Running

Now you can change exactly this process instance which you are testing. We want to change the data type of the *Begin* and *End* data elements from *String* to *Date*. Since we have data flow correctness checks we cannot directly change the data types of data elements which are connected to process steps which have assigned activities already. The data type of the activity's / service's parameters would not match the data type of the data element. In this situation we have a couple of methods to reach our goal (delete data edges, data elements, reassign activities, etc.). In this case we decide to delete the existing *Begin* and *End* data elements with type *String* and create new ones with type *Date*.

Select the *Begin* and *End* data elements and chose the *Delete Data Element* change operation in the *Change Operations* view or in the right-click menu. Insert new data elements and change their type to *Date* in the *Properties* view.

The deletion of the data elements of type String caused warnings on the process steps which were connected to these elements. In the *Check Report View* the warnings are listed in more detail. To correct the process graph double-click the *"Fill out request for time off form"* process step. Switch to the parameter wizard page from the first process step, select the *Begin* parameter and press the *Edit parameter* button. Change the data type to *Date* in the upcoming dialogue and press *OK*.



Adjust properties of node 'Fill out request for time off form'

Parameters

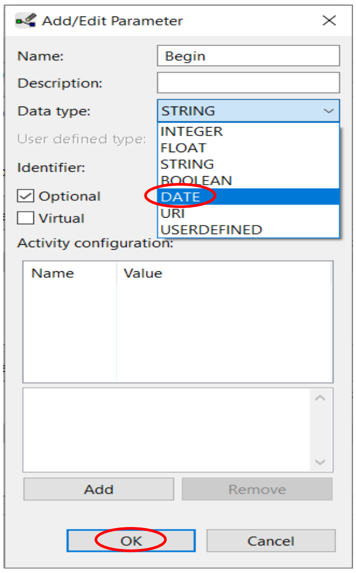
On this page you can define the input and output parameters for the activity.

Input Parameters (3)

Name	Description	Data Type	Optional	Virtual
Begin		STRING	yes	no
End		STRING	yes	no
Originator		STRING	yes	no

Output Parameters (3)

Name	Description	Data Type	Optional	Virtual
Begin		STRING	no	no
End		STRING	no	no
Originator		STRING	no	no



Add/Edit Parameter

Name: Begin

Description:

Data type: STRING

User defined type:

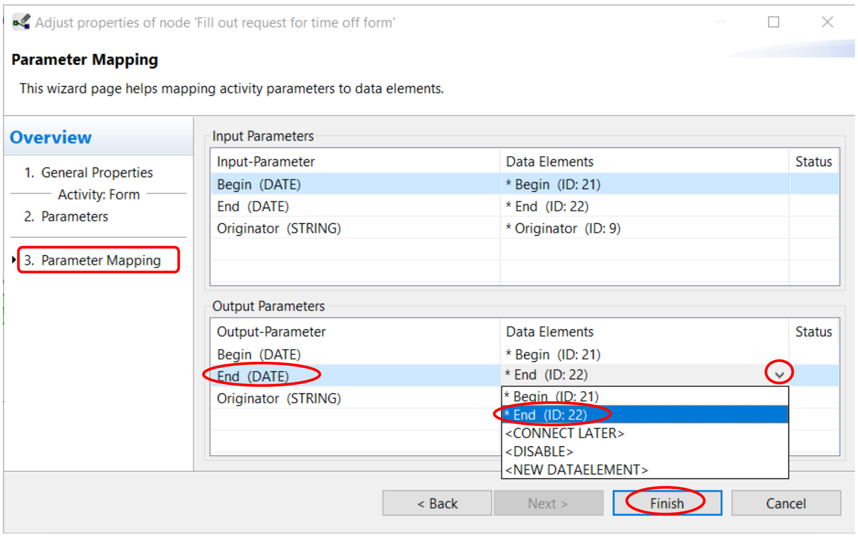
Identifier:

☑ Optional

□ Virtual

Activity configuration:

Do the same with the *End Input Parameter* and the *Begin* and *End Output Parameters*. Click *"Next >"*. Map the parameter to their corresponding data elements if it is not done automatically. The parameter mapping should look like this now:



Adjust properties of node 'Fill out request for time off form'

Parameter Mapping

This wizard page helps mapping activity parameters to data elements.

Input Parameters

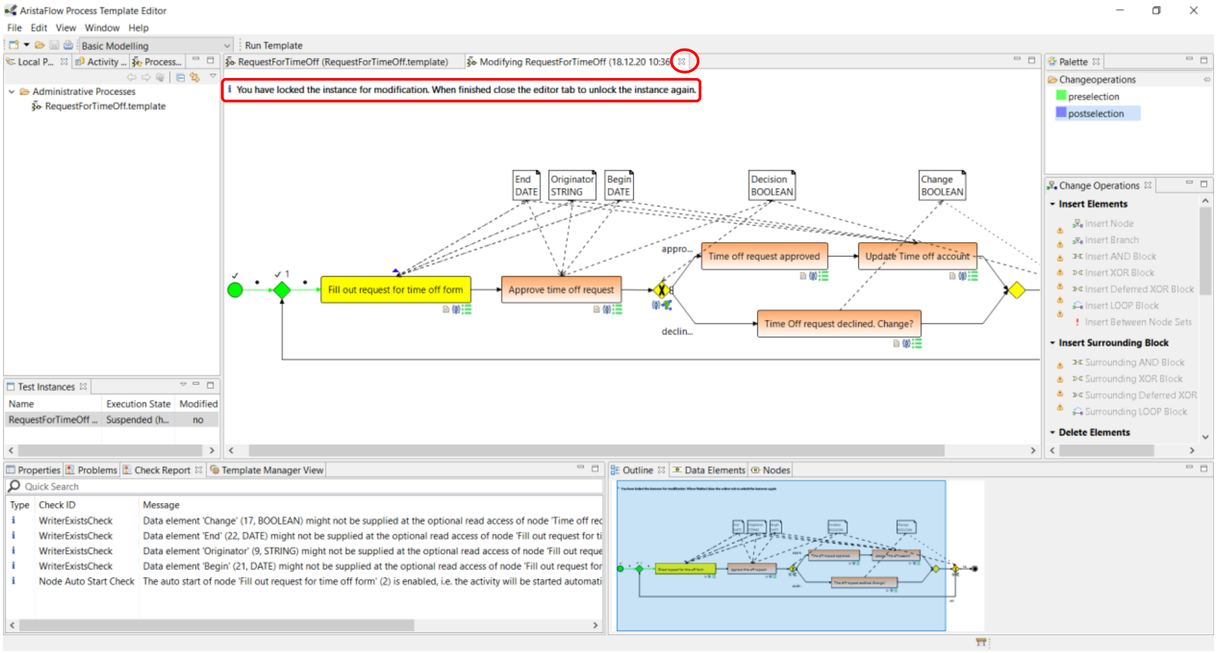
Input-Parameter	Data Elements	Status
Begin (DATE)	* Begin (ID: 21)	
End (DATE)	* End (ID: 22)	
Originator (STRING)	* Originator (ID: 9)	

Output Parameters

Output-Parameter	Data Elements	Status
Begin (DATE)	* Begin (ID: 21)	
End (DATE)	* End (ID: 22)	
Originator (STRING)		

Press the *finish* button.

Change the other process steps with a warning symbol accordingly. If you are finished the final result should not have any errors or warnings. Save the process change with Ctrl-S and press OK in the upcoming dialogue to commit the changes to the test client. Then close the modifiying editor tab from the instance. The instance were locked in the editor, as long as you edited it.



You have locked the instance for modification. When finished close the editor tab to unlock the instance again.

Test Instances

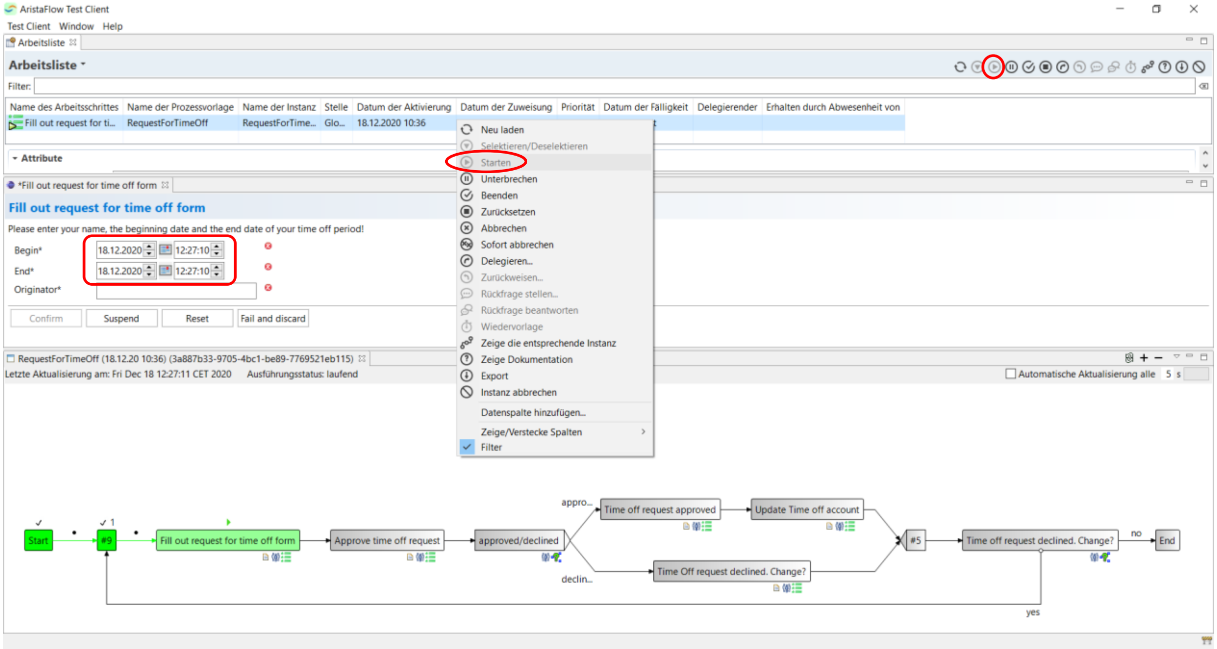
Name	Execution State	Modified
RequestForTimeOff ...	Suspended (h...	no

Properties | Problems | Check Report | Template Manager View

Quick Search

Type	Check ID	Message
i	WriterExistsCheck	Data element 'Change' (17, BOOLEAN) might not be supplied at the optional read access of node 'Time off rec
i	WriterExistsCheck	Data element 'End' (22, DATE) might not be supplied at the optional read access of node 'Fill out request for ti
i	WriterExistsCheck	Data element 'Originator' (9, STRING) might not be supplied at the optional read access of node 'Fill out reque
i	WriterExistsCheck	Data element 'Begin' (21, DATE) might not be supplied at the optional read access of node 'Fill out request for
i	Node Auto Start Check	The auto start of node 'Fill out request for time off form' (2) is enabled, i.e. the activity will be started automati

Switch back to the *Test Client* and start the *"Fill out request for time off form"* step from the worklist again by selecting *Start* in the right-click menu or in the toolbar menu. The *String* input fields are changed to nicer *Date* fields.



Arbeitsliste

Filter:

Name des Arbeitsschrittes	Name der Prozessvorlage	Name der Instanz	Stelle	Datum der Aktivierung	Datum der Zuweisung	Priorität	Datum der Fälligkeit	Delegierender	Erhalten durch Abwesenheit von
Fill out request for ti...	RequestForTimeOff	RequestForTime...	Glo...	18.12.2020 10:36					

*Fill out request for time off form

Fill out request for time off form

Please enter your name, the beginning date and the end date of your time off period!

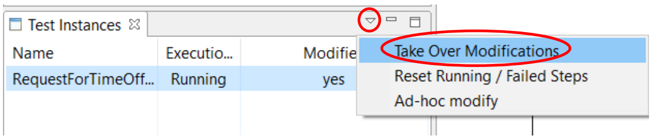
Begin*	18.12.2020 12:27:10
End*	18.12.2020 12:27:10
Originator*	

RequestForTimeOff (18.12.20 10:36) (3a887b33-9705-4bc1-be89-7769521eb115)

Letzte Aktualisierung am: Fri Dec 18 12:27:11 CET 2020 Ausführungsstatus: laufend

Enter some test values and close the form with *"Confirm"*. The process engine changes the execution state of the step to *"finished"* and activates the *"Approve time off request"* step which in turn is shown to the user in the worklist.

If you like the changes we have done, switch back to the *Process Template Editor* and select *Take Over modifications*. The changes are now propagated from the test instance of the process to the process template.



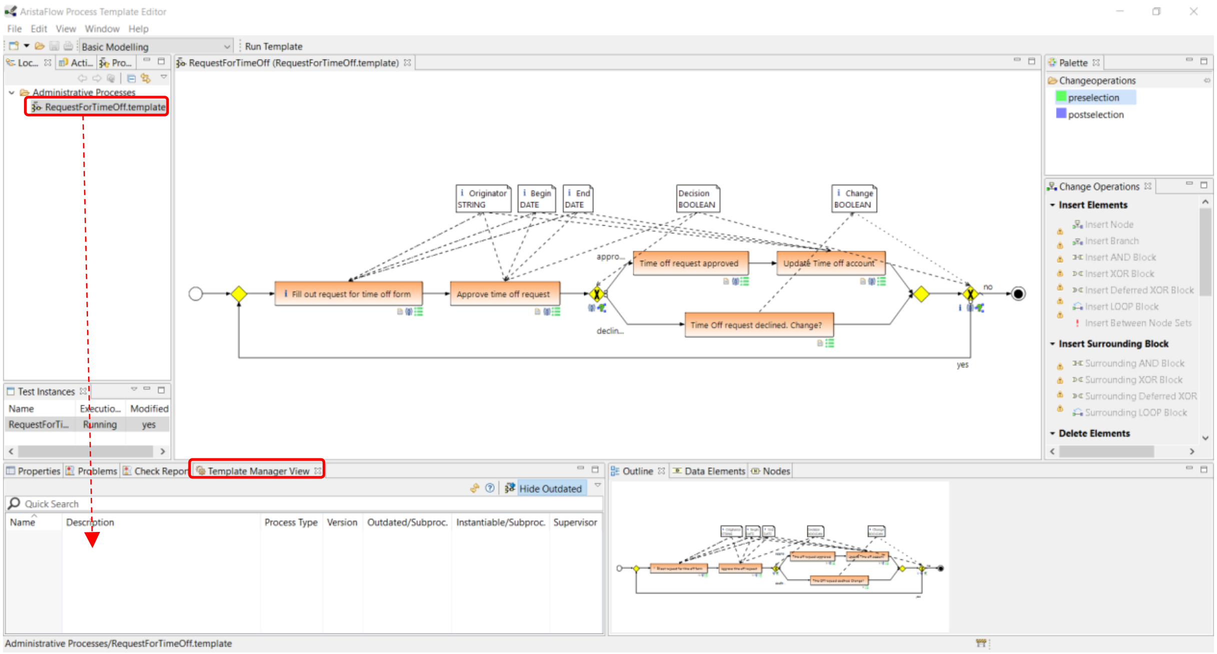
Name	Executio...	Modifie
RequestForTimeOff...	Running	yes

Go on with the testing of the process and check if the decision rules on the *"approved / declined"* and on the *"Change time off request on decline?"* process steps are correct.

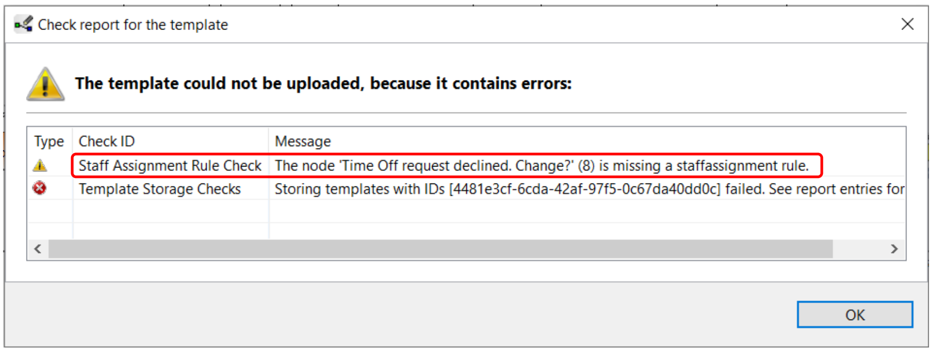
If you are satisfied with the process close the *Test Client* and switch back to the *Process Template Editor*. In the next step we deploy the process to the *AristaFlow Server* for productive operations.

Deployment of the Request for Time off process

For productive operations the *Request for Time off* process must be deployed to the AristaFlow *Server Repository*. The easiest way is to simply drag'n'drop the *"RequestForTimeOff.template"* from the *AFResourceNavigator* view to the *TemplateManagerView* at the bottom of the editor.



On deployment the process passes through a couple of further completeness and correctness checks to guarantee that the process can be run without errors. If the checks are successful the *Request for Time off* process is uploaded to the *AristaFlow Server*. If not the errors are listed and the upload is cancelled (see screenshot). In this case the Staff Assignment for the process step *"Time off request declined. Change?"* is missing and therefore the template could not be uploaded. To fix this problem, the *InstanceInitiator* is to add as Agent to this step (see page 30 Staff Assignment).

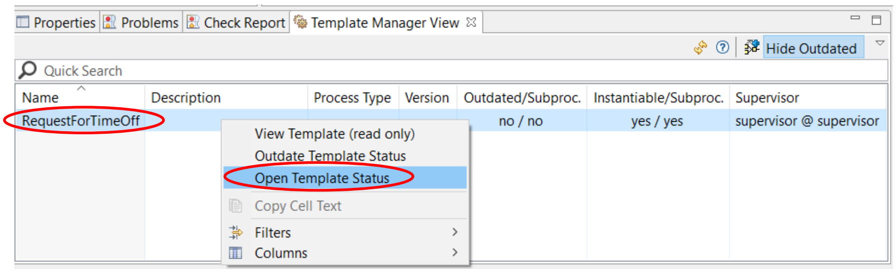

Check report for the template

The template could not be uploaded, because it contains errors:

Type	Check ID	Message
⚠	Staff Assignment Rule Check	The node 'Time Off request declined. Change?' (8) is missing a staffassignment rule.
⊗	Template Storage Checks	Storing templates with IDs [4481e3cf-6cda-42af-97f5-0c67da40dd0c] failed. See report entries for

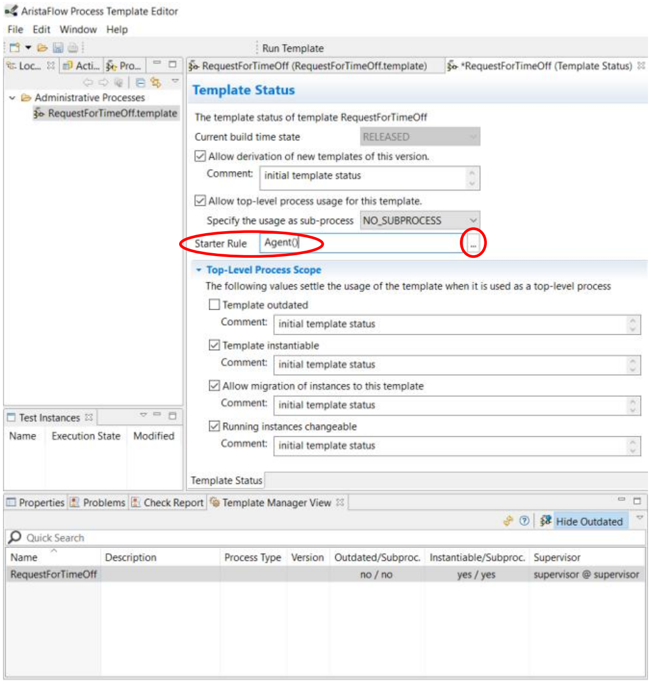
Before we can run our process we have to configure some properties, e.g. the staff which is allowed to start the process.

Right-click the *Request For Time Off* process in the *TemplateManagerView* and select *Open Template Status*.



Name	Description	Process Type	Version	Outdated/Subproc.	Instantiable/Subproc.	Supervisor
RequestForTimeOff				no / no	yes / yes	supervisor @ supervisor

Under the *Starter Rule* point you can specify the agents, roles, org. units, etc. which should be authorized to start a *Request for time off* process. You can open the *Staff Assignment* wizard with the *"…"* button next to the *Starter Rule* to assign a complex starter rule or enter an agent manually. For our demo it is sufficient to enter *Agent()* in the input field next to the *Starter Rule*, which allows everyone to start the process. Save the change with Ctrl-S and close the *Template Status*.



Template Status

The template status of template RequestForTimeOff

Current build time state: RELEASED

☑ Allow derivation of new templates of this version.
Comment: initial template status

☑ Allow top-level process usage for this template.
Specify the usage as sub-process: NO_SUBPROCESS

Starter Rule: Agent() ...

Top-Level Process Scope

The following values settle the usage of the template when it is used as a top-level process

☐ Template outdated
Comment: initial template status

☑ Template instantiable
Comment: initial template status

☑ Allow migration of instances to this template
Comment: initial template status

☑ Running instances changeable
Comment: initial template status

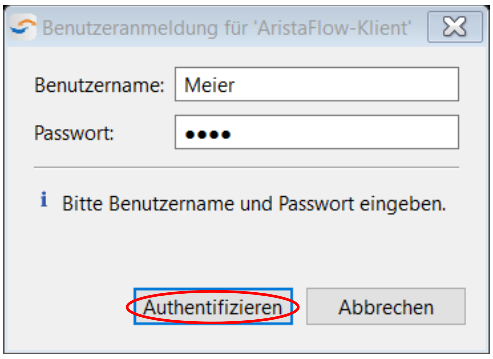
Name	Description	Process Type	Version	Outdated/Subproc.	Instantiable/Subproc.	Supervisor
RequestForTimeOff				no / no	yes / yes	supervisor @ supervisor

That's all! The *Request for Time off* process has been modeled, validated and deployed. Now we can switch to the end user clients to do a real live execution of the process.

Process Execution

Start the *AristaFlow Client* application which you can find in your Windows start menu under → *AristaFlow BPM Suite* → *Client* or in your installation folder (…\Client\Client.exe).

Authenticate with the user name *"Meier"* and password *"test"* in the upcoming login dialogue.



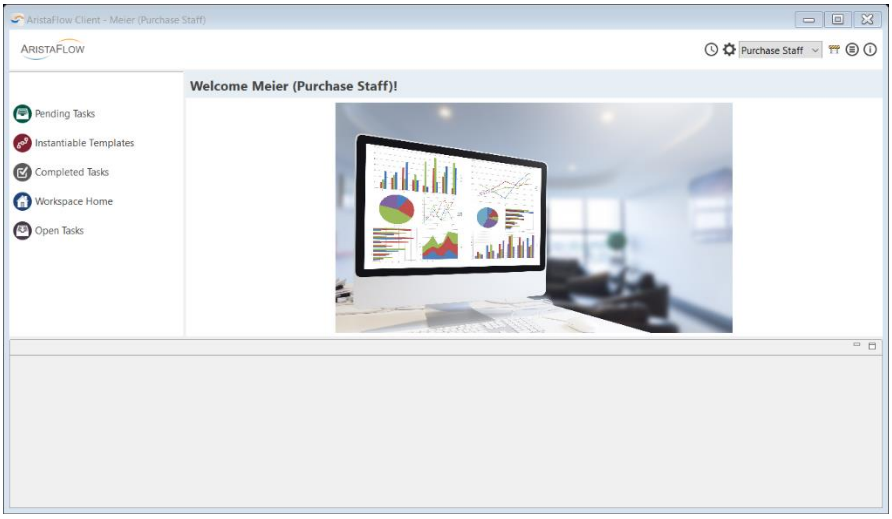
Benutzeranmeldung für 'AristaFlow-Klient'

Benutzername: Meier

Passwort: ●●●●

ℹ Bitte Benutzername und Passwort eingeben.

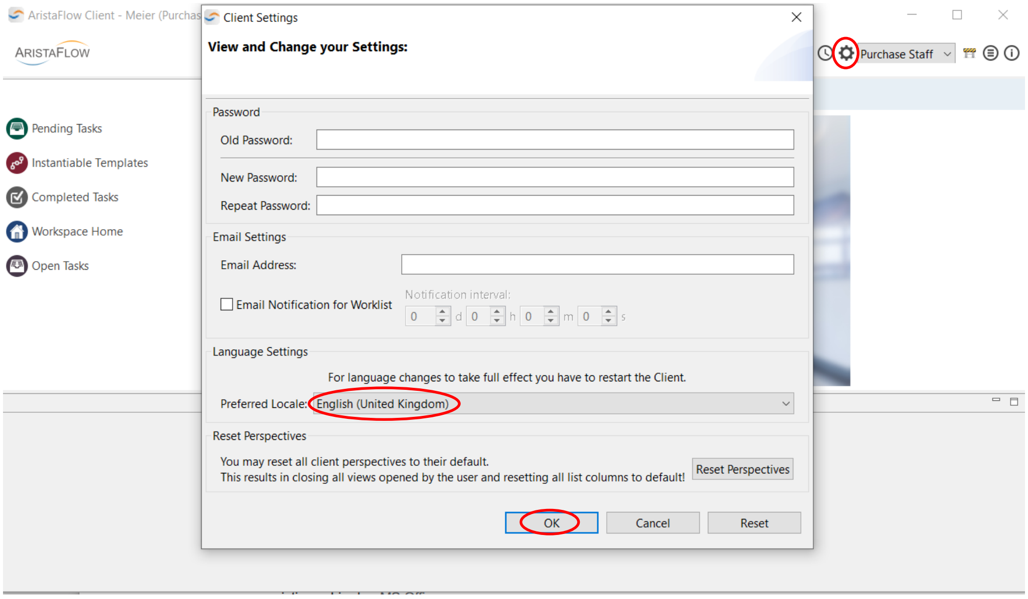
The *AristaFlow Rich Client* opens. This is the working place where the end user starts processes and executes the work which is delegated to his / her worklist.



AristaFlow Client - Meier (Purchase Staff)

Welcome Meier (Purchase Staff)!

Hint:
The client is internationalized and is initially configured to show the language which is set in the operation system. To change the language push the button left to the drop down box and choose the language for the current user (see following screenshot).



Client Settings

View and Change your Settings:

Password

Old Password:

New Password:

Repeat Password:

Email Settings

Email Address:

□ Email Notification for Worklist

Notification interval: 0 d 0 h 0 m 0 s

Language Settings

For language changes to take full effect you have to restart the Client.

Preferred Locale: English (United Kingdom)

Reset Perspectives

You may reset all client perspectives to their default.
This results in closing all views opened by the user and resetting all list columns to default!

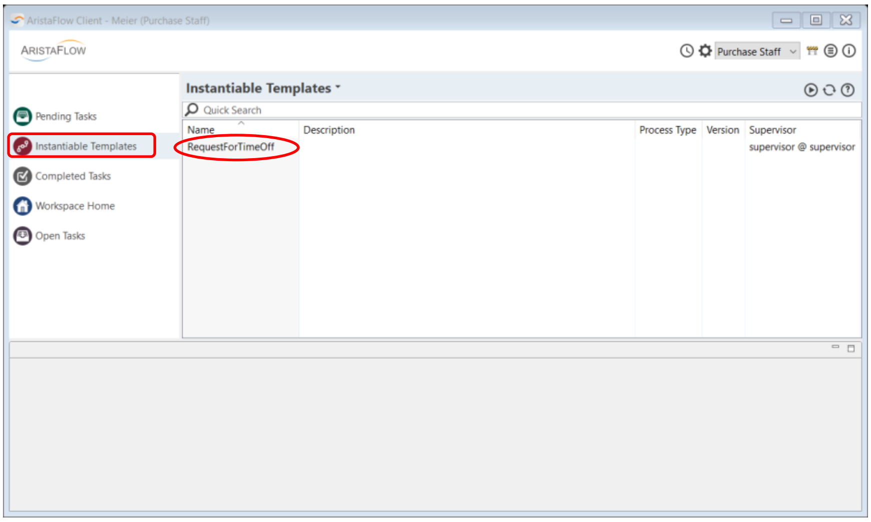
Before we can go on with our request for time off process we have to start the client which is responsible for the execution of all process steps which do not have user interaction first.

AristaFlow Automatic Client

The AristaFlow *Automatic Client* acts like a normal client with a normal worklist, except that the *Automatic Client* needs no user interaction to execute a process step. As soon as a worklist item is delegated to its worklist the *Automatic Client* executes it immediately. In our *request for time off process* the *Automatic Client* is responsible for executing the rules on the XOR and the Loop steps.

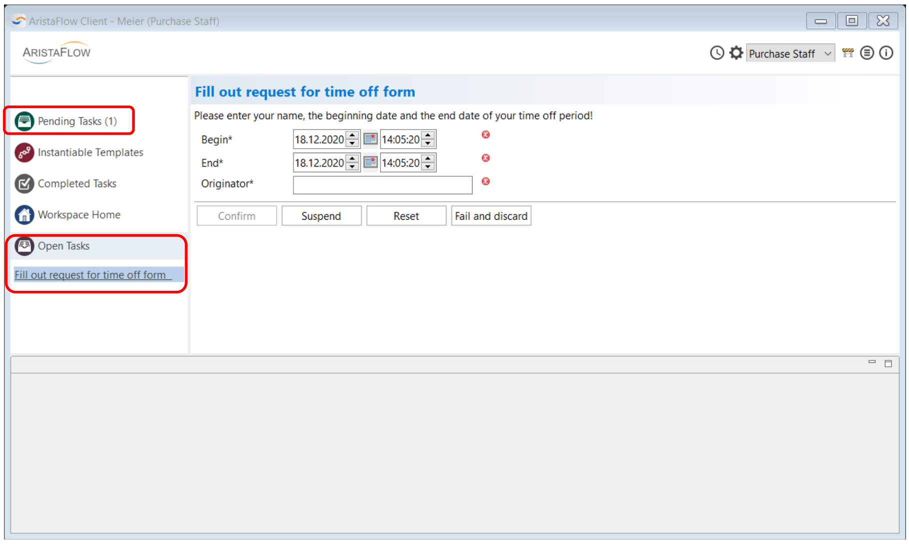
Start the AristaFlow *Automatic Client* which you can find in your Windows start menu under → AristaFlow *BPM Suite* → *Console* → *AutomaticClient with console* or in your installation folder (…\AutomaticClient\bin\automaticclient.bat). Since the *Automatic Client* is nearly used in every process it is more comfortable to install it as a Windows service and keep it running together with the AristaFlow Server. If you are done, switch back to the *Client*.

Go to the *Instantiable Templates* view using the hyperlink in the workspace view on the left and double-click the *RequestForTimeOff* process to start an instance of this process template.



Name	Description	Process Type	Version	Supervisor
RequestForTimeOff				supervisor @ supervisor

If you have activated the autostart property for this step in the *Process Template Editor* the first process step is started automatically. Otherwise switch to the *Pending Tasks* and double-click the related worklist item.



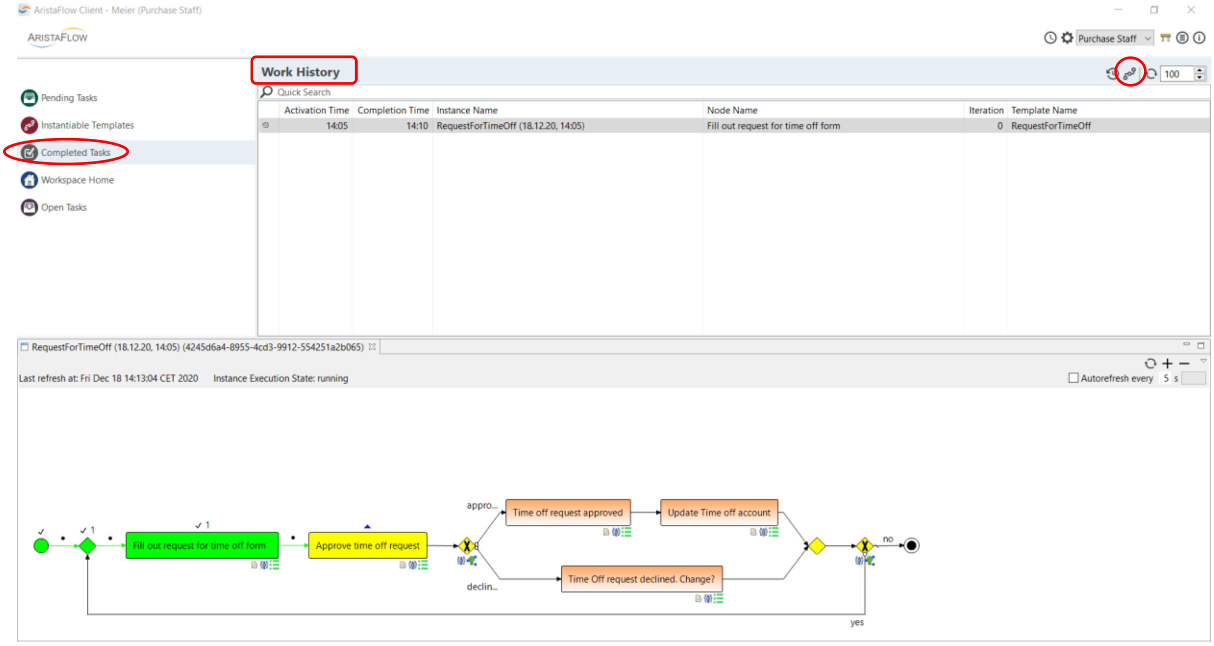
Fill out request for time off form

Please enter your name, the beginning date and the end date of your time off period!

Begin* 18.12.2020 14:05:20
End* 18.12.2020 14:05:20
Originator*

Select a begin and an end date and enter "*Meier*" in the Originator text field. If you entered valid values the red dots next to the fields disappear and you can finish the form with the *Confirm* button. Alternatively if you are not exactly sure about the begin and end date you can use the "*Suspend*" button which causes the form to store the actually entered values and delegates the item back to your worklist. If you are sure start the step again and complete it. The form will open automatically. Otherwise you will find the individual started activities (forms) under the hyperlink *Open Tasks*.

Now the process step is disappeared and the request for time off has been send to the supervisor of the employee "*Meier*". This can be monitored by switching to the *Completed Tasks* view, selecting the step which we have just completed, and pressing the *process chains* icon in the upper right corner.



AristaFlow Client - Meier (Purchase Staff)

Work History

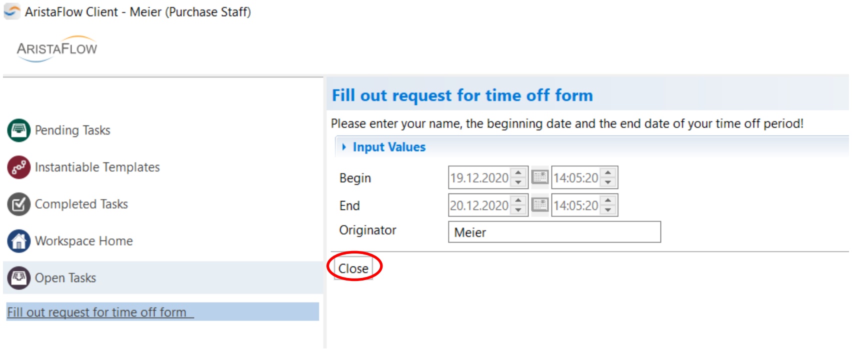
Activation Time	Completion Time	Instance Name	Node Name	Iteration	Template Name
14:05	14:10	RequestForTimeOff (18.12.20, 14:05)	Fill out request for time off form	0	RequestForTimeOff

RequestForTimeOff (18.12.20, 14:05) (4245d6a4-8955-4cd3-9912-554251a2b065)

Last refresh at: Fri Dec 18 14:13:04 CET 2020   Instance Execution State: running

☐ Autorefresh every 5 s

Additionally you can re-open completed steps in read-only mode by double-clicking entries in the *Work History* view. This is useful if you often have many process instances of the same type and you want to keep your data input consistent (see the following screenshot for a completed form).



AristaFlow Client - Meier (Purchase Staff)

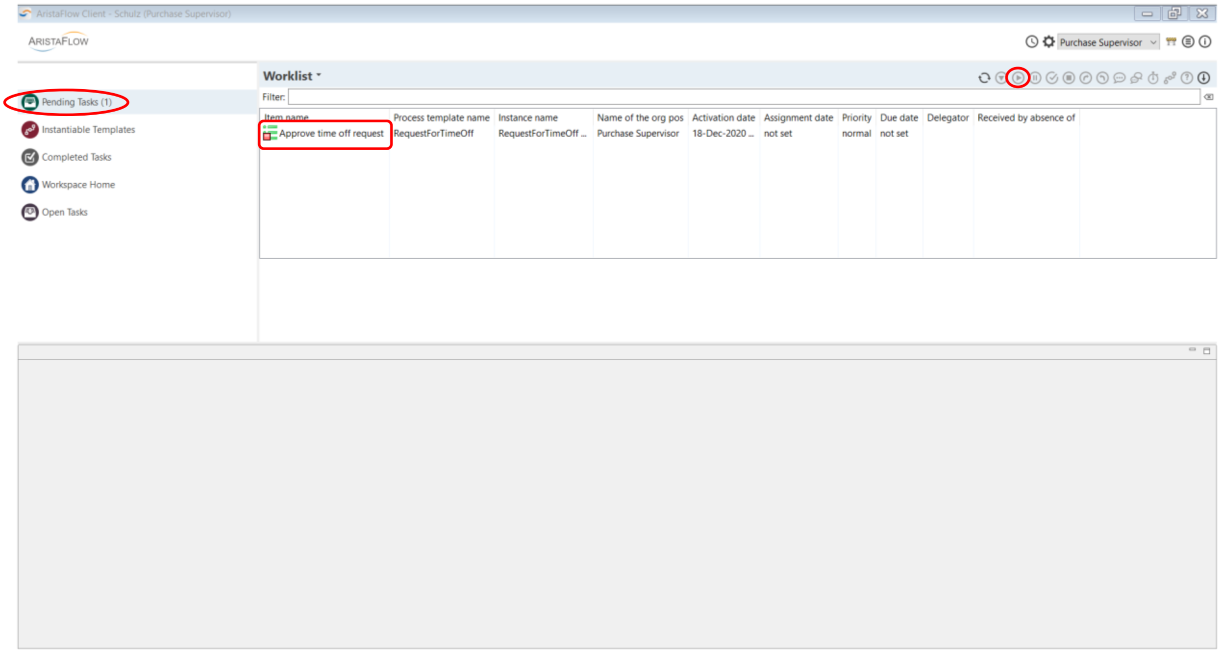
Fill out request for time off form

Please enter your name, the beginning date and the end date of your time off period!

Input Values

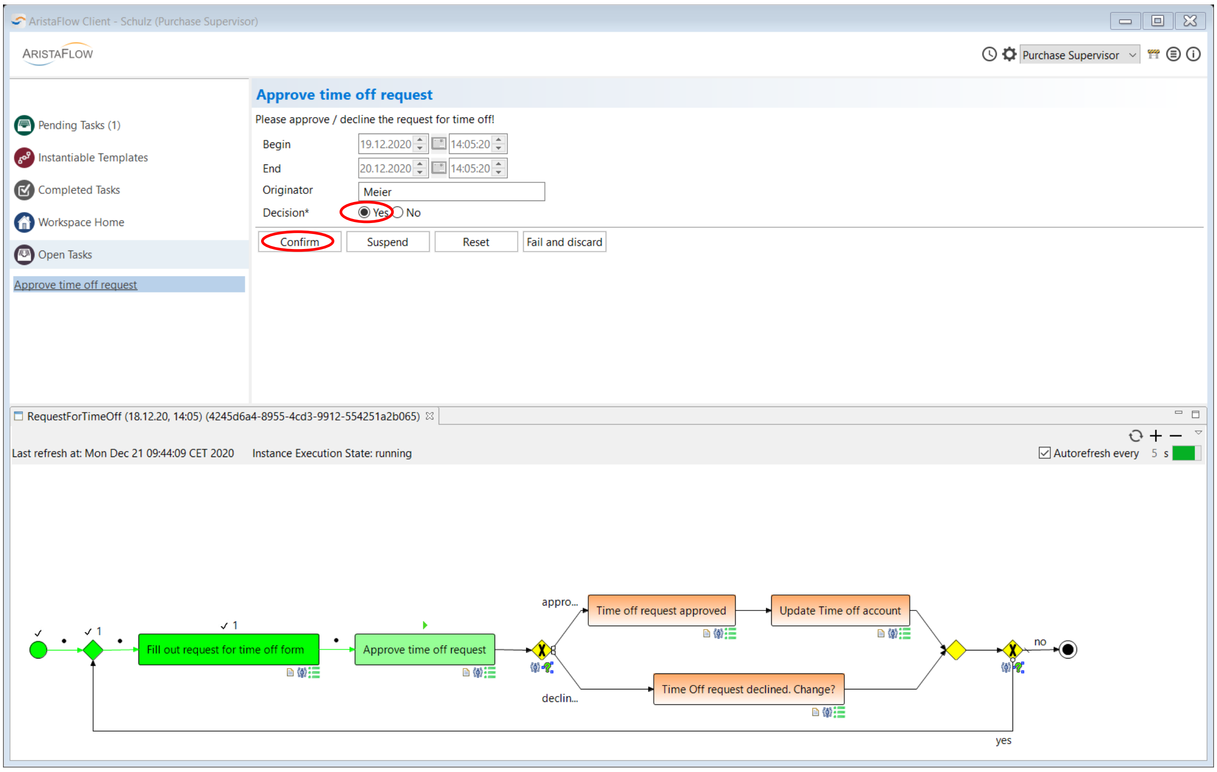
Begin	19.12.2020	14:05:20
End	20.12.2020	14:05:20
Originator	Meier	

Our *Request for time* off should be in the "*Approval*" step which is to be executed by the supervisor of the employee "*Meier*" which is Mr. "*Schulz*". To get to his worklist we have to open another AristaFlow Rich Client from the Windows menu and log-in with *Schulz* / *test*.



Item name	Process template name	Instance name	Name of the org pos	Activation date	Assignment date	Priority	Due date	Delegator	Received by absence of
Approve time off request	RequestForTimeOff	RequestForTimeOff ...	Purchase Supervisor	18-Dec-2020 ...	not set	normal	not set		

The hyperlink *Pending Tasks* of the logged-in users show exactly those worklist items that are to be carried out by the respective users. After a click on that you can see the “*Approve time off request*” step, exactly how it is modeled in the process template. Double-click the worklist entry or press the start icon in the toolbar on the right hand side to open the “*Approve time off request*” step.



Approve time off request

Please approve / decline the request for time off!

Begin 19.12.2020 14:05:20

End 20.12.2020 14:05:20

Originator Meier

Decision* ● Yes ○ No

Confirm | Suspend | Reset | Fail and discard

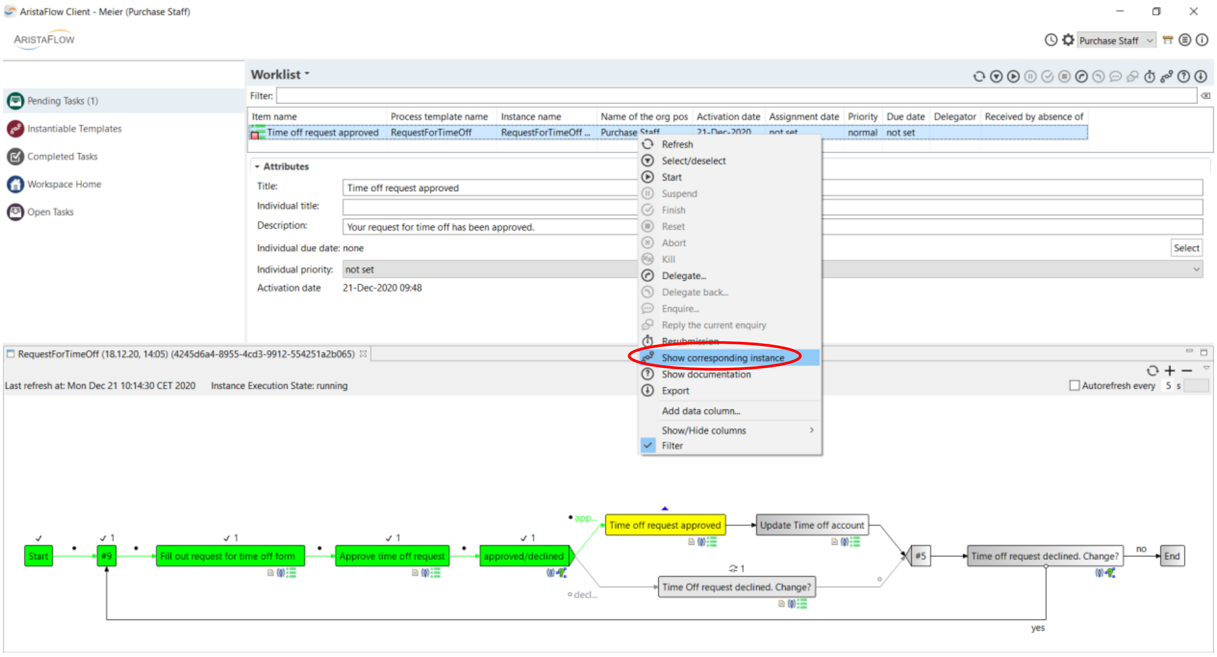
RequestForTimeOff (18.12.20, 14:05) (4245d6a4-8955-4cd3-9912-554251a2b065)

Last refresh at: Mon Dec 21 09:44:09 CET 2020 Instance Execution State: running

☑ Autorefresh every 5 s

Fill out request for time off form
Approve time off request
appro...
Time off request approved
Update Time off account
Time Off request declined. Change?
declin...
no
yes

In this case we approve the request and close the form with *Confirm*.
The process passes the decision step (XOR) which is executed by the automatic client and fills the worklist of the request for time off originator with the approval information step. To follow the further execution switch back to the originator *Meier*.



AristaFlow Client - Meier (Purchase Staff)

Worklist

Filter:

Item name	Process template name	Instance name	Name of the org pos	Activation date	Assignment date	Priority	Due date	Delegator	Received by absence of
Time off request approved	RequestForTimeOff	RequestForTimeOff ...	Purchase Staff	21-Dec-2020	not set	normal	not set		

Attributes

Title: Time off request approved

Individual title:

Description: Your request for time off has been approved.

Individual due date: none

Individual priority: not set

Activation date 21-Dec-2020 09:48

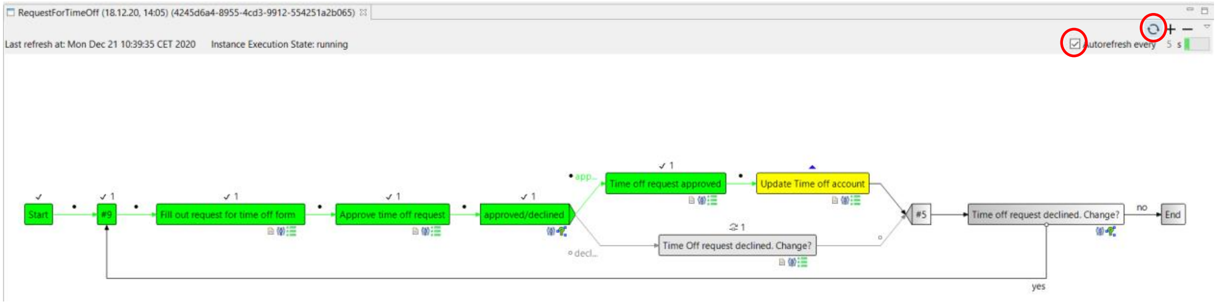
RequestForTimeOff (18.12.20, 14:05) (4245d6a4-8955-4cd3-9912-554251a2b065)

Last refresh at: Mon Dec 21 10:14:30 CET 2020 Instance Execution State: running

□ Autorefresh every 5 s

Hint:
To show the instance execution click on the *process chains icon* in the upper right corner or select "show corresponding instance" in the right-click menu from the process step.

Execute the worklist item as usual and follow the instance execution in the real time monitoring view. Refresh the view automatically by ticking the box or manually by clicking the *refresh icon* upper on the right of the monitoring view. Now the process instance is in the "*Update Time off account*" step which is to be executed by the "*Human Resource Staff*".



RequestForTimeOff (18.12.20, 14:05) (4245d6a4-8955-4cd3-9912-554251a2b065)

Last refresh at: Mon Dec 21 10:39:35 CET 2020 Instance Execution State: running

☑ Autorefresh every 5 s

Start
#9
Fill out request for time off form
Approve time off request
approved/declined
app...
decl...
Time off request approved
Update Time off account
Time Off request declined. Change?
#5
Time off request declined. Change?
no
End
yes

Hint:
The finished process steps are shown in green, the current running state turns light green and the activated state turns yellow.

Apart from the other steps which we have executed with the *AristaFlow Rich Client* we want to execute this process step with the *AristaFlow Web Client*.

AristaFlow Web Client

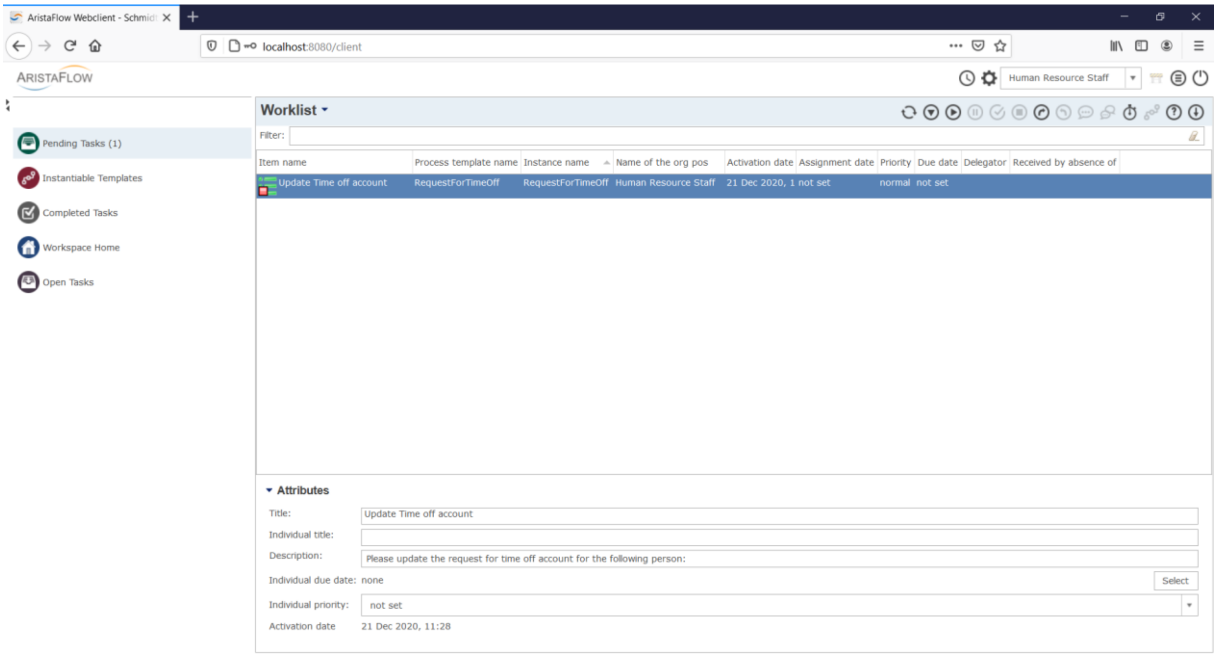
The AristaFlow Web Client offers the same look and feel and functionality as the Rich Client. Since you have nothing to install on client side it might be the favorite choice for productive environments.

To be able to start the *Web Client* in a browser we first have to start the *AristaFlow Web Server from the* Windows start menu …→ *AristaFlow BPM Suite* → *Web server with console*.

Or you find it in your installation folder under
…→ *AristaFlow BPM Suite* → *WebServer* → *bin* → *webclient.bat*.

If you mainly want to use the *Web Client* it is more comfortable to install the *Web Server* as a Windows service and keep it running together with the AristaFlow Server.

Open a browser and enter the URL http://localhost:8080/client. Log-in with Schmidt / test.

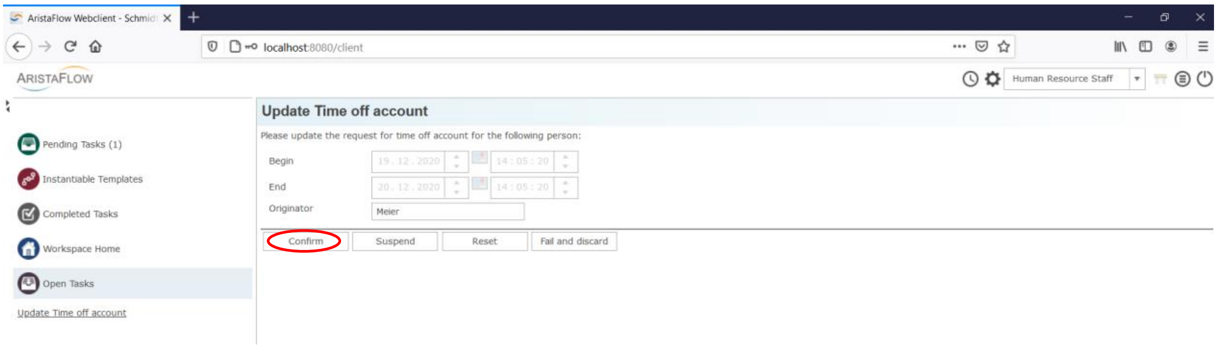


Item name	Process template name	Instance name	Name of the org pos	Activation date	Assignment date	Priority	Due date	Delegator	Received by absence of
Update Time off account	RequestForTimeOff	RequestForTimeOff	Human Resource Staff	21 Dec 2020, 1	not set	normal	not set		

Attributes

Title: Update Time off account
Individual title:
Description: Please update the request for time off account for the following person:
Individual due date: none
Individual priority: not set
Activation date 21 Dec 2020, 11:28

Execute the step by double-clicking the worklist item.



Update Time off account

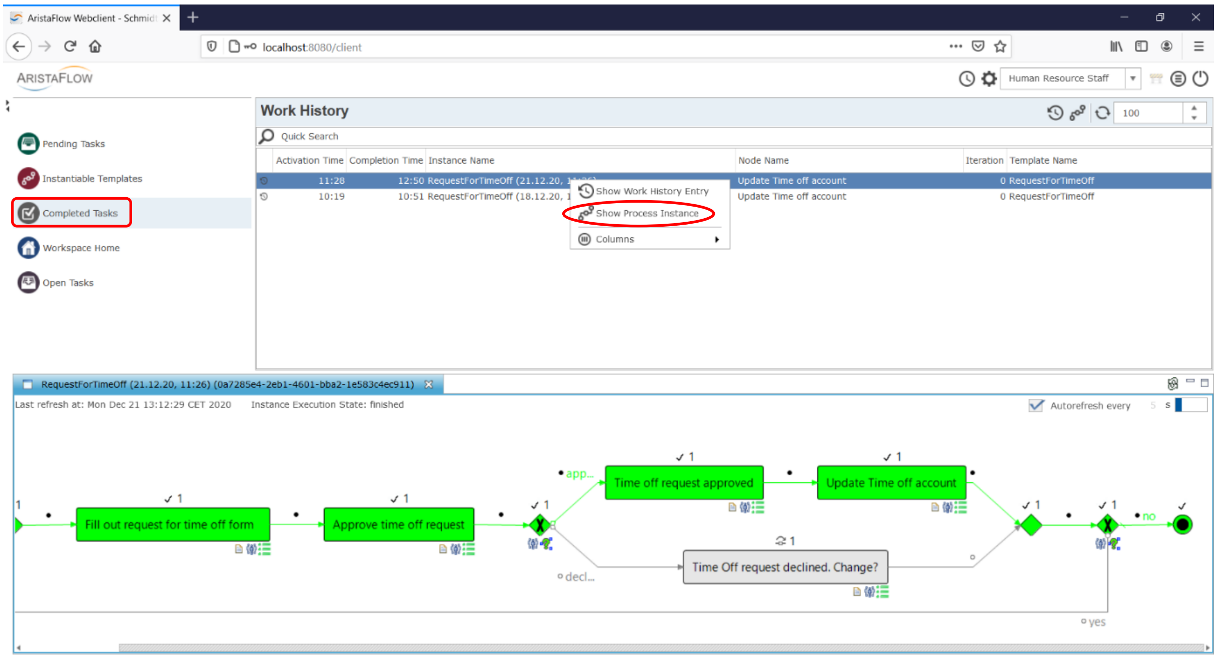
Please update the request for time off account for the following person:

Begin 19 . 12 . 2020 14 : 05 : 20
End 20 . 12 . 2020 14 : 05 : 20
Originator Meier

Confirm the form and the *"Request for Time off"* process is finished.

To be able to display the process execution status, you also need the server *Process image renderer with console*. You start it from the Windows start menu under
…→ *AristaFlow BPM Suite* → *Process image renderer with console* or in your installation folder under
…*AristaFlow-Chronos**ProcessImageRenderer**bin**processimagerenderer.bat.*

Under the hyperlink *Completed Tasks* you can find the activities that have already completed. Select the finished worklist item to be displayed from the list *Work history* and click on *Show Process Instance* in the right-click menu. The finished graph appears below the window.



	Activation Time	Completion Time	Instance Name	Node Name	Iteration	Template Name
	11:28	12:50	RequestForTimeOff (21.12.20, 1	Update Time off account	0	RequestForTimeOff
	10:19	10:51	RequestForTimeOff (18.12.20, 1	Update Time off account	0	RequestForTimeOff

RequestForTimeOff (21.12.20, 11:26) (0a7285e4-2eb1-4601-bba2-1e583c4ec911)

Last refresh at: Mon Dec 21 13:12:29 CET 2020   Instance Execution State: finished

Hint:

Since only one person can log in to the Web Client with a user profile of an Internet browser, you either need several browsers or several profiles in one browser. You can use three browsers for the three agents involved in our *"request for time off"* process, e.g. Take Firefox, Opera and Internet Explorer.

To create several user profiles, for example in the Firefox browser, proceed as follows:

- Type the command *firefox.exe –p* in the Windows start menu.
- Or alternatively type *about:profiles* into the browser URL search bar from Firefox.
- In the upcoming profile manager dialogue or about profiles page you can create additional profiles or rename, delete and start existing ones. Usually Firefox automatically creates a profile with the name "default". Now we also create a separate profile for each agent (here originator, supervisor and staff) and then log in these with a separate browser instance to the WebClient.

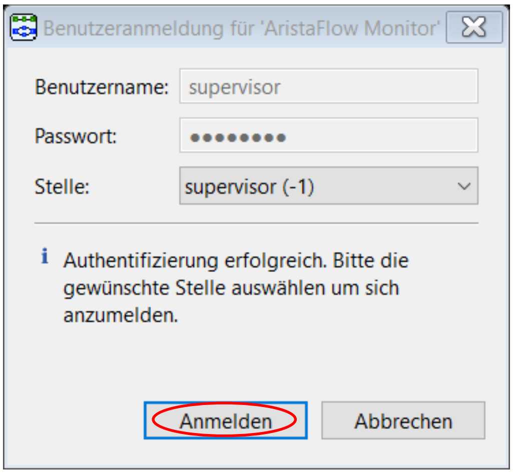
If you want to play a little more with all agents in the WebClient simply start a couple of new *"Request for time off"* processes and check what happens if you decline the time off request. But before starting Firefox, you have to open the Profile Manager as described above and select the appropriate profile. Then enter the URL http://localhost:8080/client and log in the relevant user with the password.If your done go on with the next section.

Monitor and Analyze

In addition to the simple real time monitoring capabilities in the AristaFlow *Rich Client* the AristaFlow *BPM Suite* offers a much more powerful monitoring and analysis tool: the AristaFlow *Monitor*.

Start the AristaFlow *Monitor* which you can find in your Windows start menu under … → AristaFlow *BPM Suite* → *Monitor* or in your installation folder (…\Monitor\AristaFlowMonitor.exe).

Authenticate with a valid user name and password in the upcoming login dialogue (initially *supervisor / password*).



Benutzeranmeldung für 'AristaFlow Monitor'

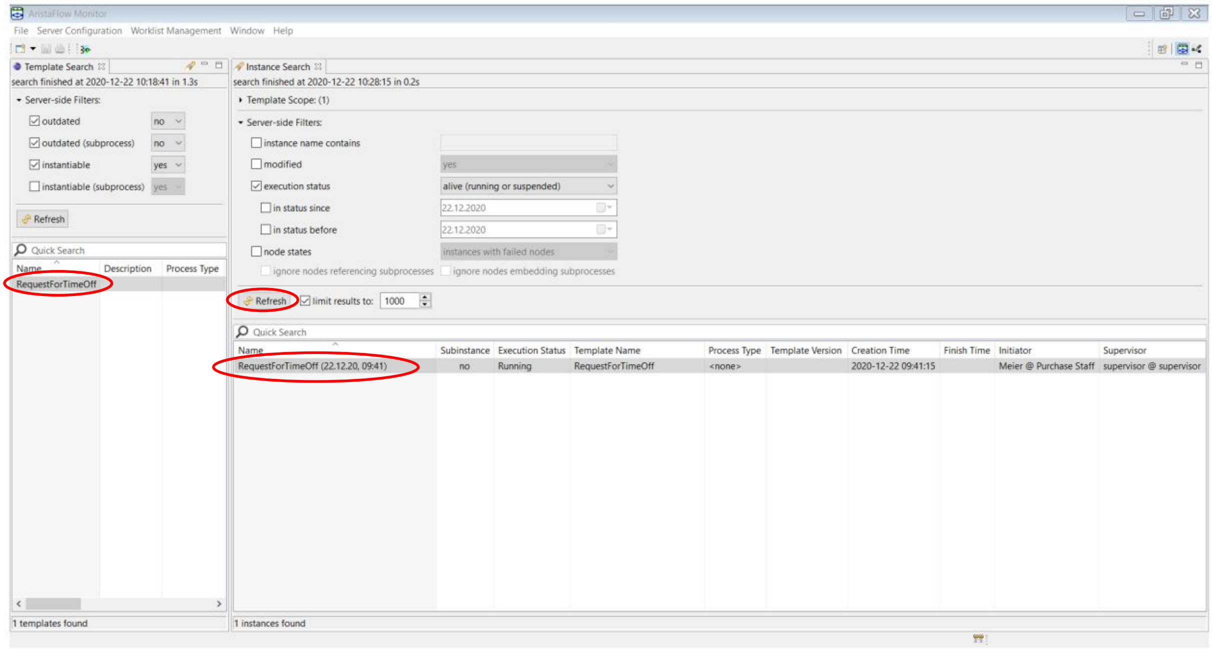
Benutzername: supervisor

Passwort: ●●●●●●●●

Stelle: supervisor (-1)

Authentifizierung erfolgreich. Bitte die gewünschte Stelle auswählen um sich anzumelden.

On the left side you can search and filter for specific process templates. Select the *Request for time off* process and double-click it. A new view *Instance Search* will open. Click on Refresh at the bottom of this view. All running or suspended instances of this process template will be listed in the workspace.



AristaFlow Monitor

Template Search

search finished at 2020-12-22 10:18:41 in 1.3s

Server-side Filters:

- ☑ outdated: no
- ☑ outdated (subprocess): no
- ☑ instantiable: yes
- ☐ instantiable (subprocess): yes

Name	Description	Process Type
RequestForTimeOff		

1 templates found

Instance Search

search finished at 2020-12-22 10:28:15 in 0.2s

Template Scope: (1)

Server-side Filters:

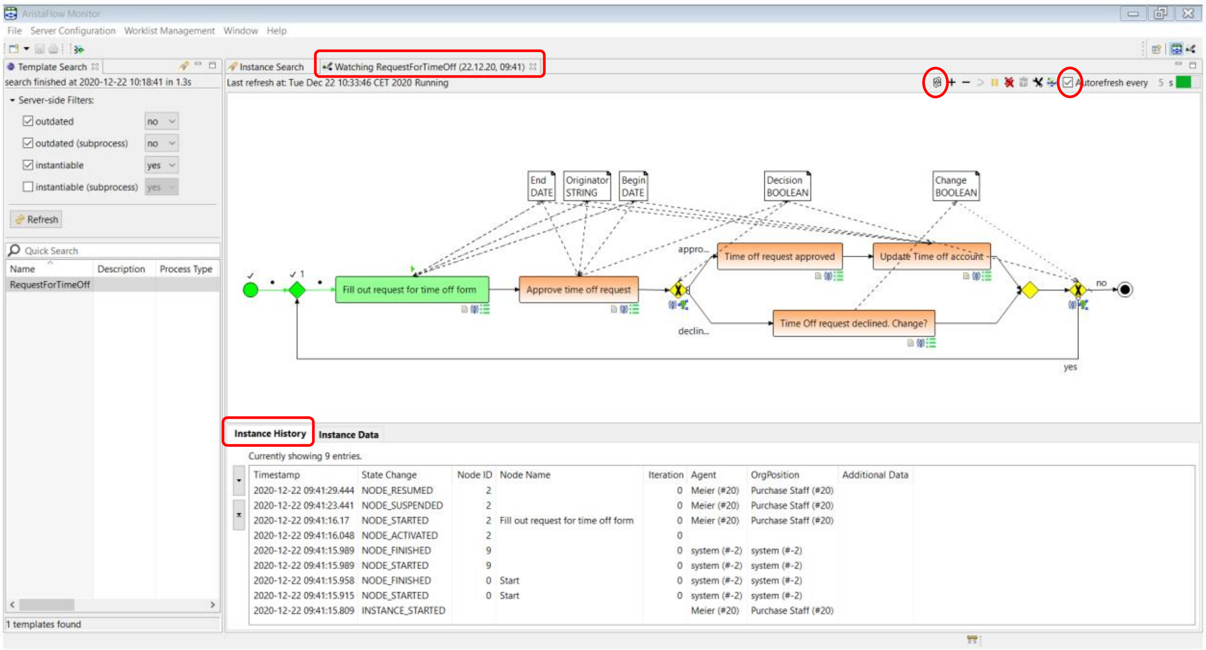
- ☐ instance name contains
- ☐ modified: yes
- ☑ execution status: alive (running or suspended)
- ☐ in status since: 22.12.2020
- ☐ in status before: 22.12.2020
- ☐ node states: instances with failed nodes
- ☐ ignore nodes referencing subprocesses ☐ ignore nodes embedding subprocesses

☑ limit results to: 1000

Name	Subinstance	Execution Status	Template Name	Process Type	Template Version	Creation Time	Finish Time	Initiator	Supervisor
RequestForTimeOff (22.12.20, 09:41)	no	Running	RequestForTimeOff	<none>		2020-12-22 09:41:15		Meier @ Purchase Staff	supervisor @ supervisor

1 instances found

Double-click one of the process instances to show the process including its execution state and detailed audit trail (*Instance History*).



search finished at 2020-12-22 10:18:41 in 1.3s

Last refresh at: Tue Dec 22 10:33:46 CET 2020 Running

Server-side Filters:

- ☑ outdated: no
- ☑ outdated (subprocess): no
- ☑ instantiable: yes
- ☐ instantiable (subprocess): yes

Currently showing 9 entries.

Timestamp	State Change	Node ID	Node Name	Iteration	Agent	OrgPosition	Additional Data
2020-12-22 09:41:29.444	NODE_RESUMED	2		0	Meier (#20)	Purchase Staff (#20)	
2020-12-22 09:41:23.441	NODE_SUSPENDED	2		0	Meier (#20)	Purchase Staff (#20)	
2020-12-22 09:41:16.17	NODE_STARTED	2	Fill out request for time off form	0	Meier (#20)	Purchase Staff (#20)	
2020-12-22 09:41:16.048	NODE_ACTIVATED	2		0			
2020-12-22 09:41:15.989	NODE_FINISHED	9		0	system (#-2)	system (#-2)	
2020-12-22 09:41:15.989	NODE_STARTED	9		0	system (#-2)	system (#-2)	
2020-12-22 09:41:15.958	NODE_FINISHED	0	Start	0	system (#-2)	system (#-2)	
2020-12-22 09:41:15.915	NODE_STARTED	0	Start	0	system (#-2)	system (#-2)	
2020-12-22 09:41:15.809	INSTANCE_STARTED				Meier (#20)	Purchase Staff (#20)	

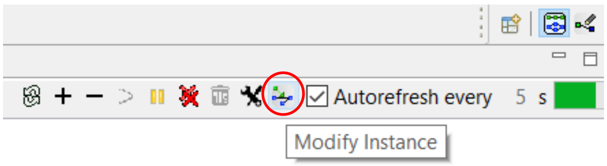
Use the Refresh button from the tool bar to synchronize with the server manually or specify an auto refresh interval in the upper left corner.

Apart from the monitoring capabilities the *Monitor* offers a couple of administrative functions:

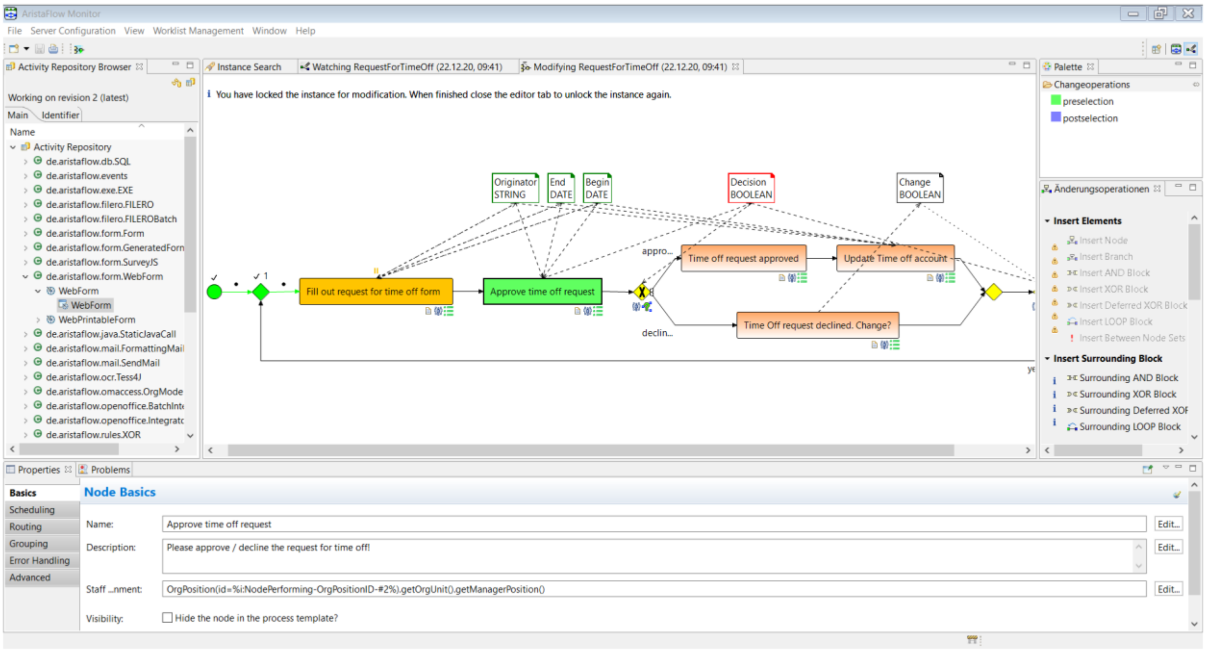
To re-open completed steps in read-only mode right-click the corresponding process step and select *"Re-open EBP (read-only)"*.

Analogously reset failed steps by right-clicking the specific step and choose *"Reset…"*.

To modify an already running instance select the *"Modify instance"* button from the toolbar.



The perspective is switched to instance change mode. Here you can adapt your process instance to changed conditions or unforeseen events. No matter what you are changing the correctness of the process is preserved by the *AristaFlow BPM Suite* even in this situation. But you can change only from the process step that has not yet started. So in our case from the second step "*Approve time off request"* you can make changes e.g. insert a *webform* instead of *userform* or insert a new node etc.



You have locked the instance for modification. When finished close the editor tab to unlock the instance again.

Node Basics

Name: Approve time off request

Description: Please approve / decline the request for time off!

Staff ...nment: OrgPosition(id=%i:NodePerforming-OrgPositionID-#2%).getOrgUnit().getManagerPosition()

Visibility: □ Hide the node in the process template?

With the powerful logging of all process execution and process change events you are able to run highly qualitative analysis to get information about bottlenecks, unused process paths and opportunities on how to optimize your process.

Epilogue

With the monitoring and analyzing capabilities of the *AristaFlow Monitor* we closed the process lifecycle and we are finished with the overview tutorial.

Note that all functions which were demonstrated as well as those which were not demonstrated are also available as a powerful Server-API. This enables you to create your own client applications or to extend ours with further functionality.

We hope you enjoyed the trip through some of the functions of our AristaFlow BPM Suite.

Contact us to learn more…

Optimize the "Request for Time off" process

Ok, you're still not satisfied with your *Request for Time off* process. Contact us and we will show you some more features to optimize the process.

The following optimizations will we explained step by step:

- The name of the originator should be added to the name of the worklist items to get a better separation of different request for time off originators.
- To enter your own name in a *"Request for Time off"* form is ugly. The name of the originator should automatically calculates and send to the form of the supervisor.
- The time off account should be updated automatically using a database component.
- The human resource staff should only inspect the automatic update of the time off account.