Use Cases

Orchestrator · https://extensionkit.unitfly.com/documentation/orchestrator-use-cases/

Calculate the value of the payment for kilometers an employee travels

A user wants to calculate the value of the payment for the kilometers an employee traveled, based on his specific km rate. The employee has to fill the form, providing an amount of km traveled separately for the city and highway cycle.

There are 3 properties which have to be calculated in the order as follows:

1. Total km traveled – *sum of km in the city and highway cycle calculated by Extended Property Calculations*
2. Driver specific km rate – *Extended Property Operation rule finds the proper rate for the driver*
3. Total settlement payment – *multiplying total km traveled and driver specific km rate with Extended Property Calculations*

Using the Orchestrator module, the user invokes the rules from Extended Property Calculations and Extended Property Operations and orchestrates the chain of actions. All properties are calculated in one cycle.

Check the value of order and invoice

A user receives an invoice that is related to an order, and uses the Orchestrator to check if the value on the invoice matches the value of the order. And then, depending on the results from the Orchestrator, the user calls the orchestrated rules in other modules.

- If values are the same, the confirmation is not needed and the invoice is passed to the accounting.
- If the value on the invoice is different that the one on the order, then the invoice goes through the confirmation and an email is sent to the responsible person to confirm the invoice.

Name	Compare prices
Description	Compare price on Invoice with the price on Order
Enabled	Yes (default)
Configuration mode	Simple
Trigger type	Object entered state
▷ Monitored properties	(not set)
◢ Trigger source	
Object Type	Document {53F0C8FD-0BF0-47C4-8FA6-4C2D0DADB650}
Class	Invoice CL.Invoice
Workflow	Workflow (not set)
State	State (not set)
▷ Search filters	(not set)
Execution mode	EventHandler (default)
◢ Orchestration conditions	

Orchestration conditions	
Basic conditions (1)	1 Item(s)
Basic condition [1]	
Condition for	Source
Search conditions	Order is not empty
Add Basic condition	
Math conditions (1)	1 Item(s)
Price	
Condition for	Source
Dynamic values from	Source
Property	Price PD.Price
Operator	=
Value type	SimpleValue
Value	{Order.Price}
Add Math condition	
Execute when conditions are satisfied (1)	1 Item(s)
SendInvoiceToAccounting	
RuleModule	Extended property operations
Rule alias	SendInvoiceToAccounting
On error	Continue (default)
Rule source	Source (default)
Add Orchestrated rule	
Execute when conditions are not satisfied (1)	1 Item(s)
SendMailToCeo	
RuleModule	Extended e-mail notifications
Rule alias	SendMailToCeo
On error	Continue (default)
Rule source	Source (default)