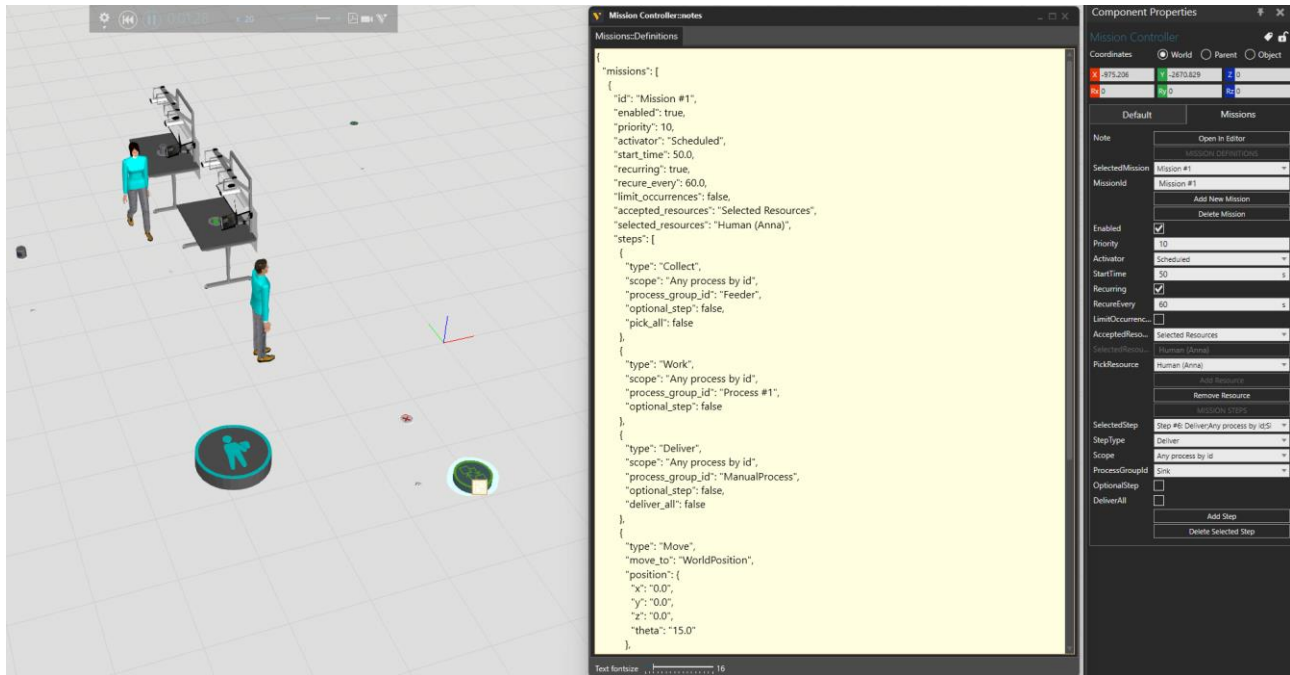


Process Modeling Missions Reference Guide

Visual Components 4.7 | Version: June 1st, 2023



Missions is a feature of Process Modeling (PM) library that allows you to define resource's task flow. A mission consists of steps that will be executed by a PM resource in the defined order.

This document includes the following topics:

- Capabilities
- Compatible components
- Limitations
- How to define a mission
- Activators and steps
- Note editor
- Debugging and troubleshooting

Contents

- Terminology 3
- Capabilities 4
- Compatibility 4
- Limitations 5
- How to use 6
- Defining missions 7
 - Mission Definitions 7
 - Adding, Removing and Renaming Missions 7
 - Activators 8
 - Accepted Resources 9
 - Mission Types 9
- Note Editor 11
- Troubleshooting 12

Terminology

Mission.	Refers typically to Mission instance, but can also mean mission definition in some contexts.
Mission controller.	User interface to define mission definitions and place holder to contain them. This data is sent to the connected transport controller on the simulation start in JSON format.
Mission definition.	Definition of a mission from which mission instances can be created into the simulation. Contains all information needed to create the mission instances of this type.
Mission instance.	One occurrence of the mission created during the simulation and defined by the mission definition.
Process Modeling (PM).	The Process Modeling is a feature provided by the Visual Components software that allows you to manage products, processes and process flows in a layout.
(Mission) steps.	Task list executed by the resource in the defined order.
Submission.	Set of steps defined by another mission definition that are called from the “parent” mission or another submission. Any mission, except the parent mission can be called regardless of whether the mission is enabled or not.

Capabilities

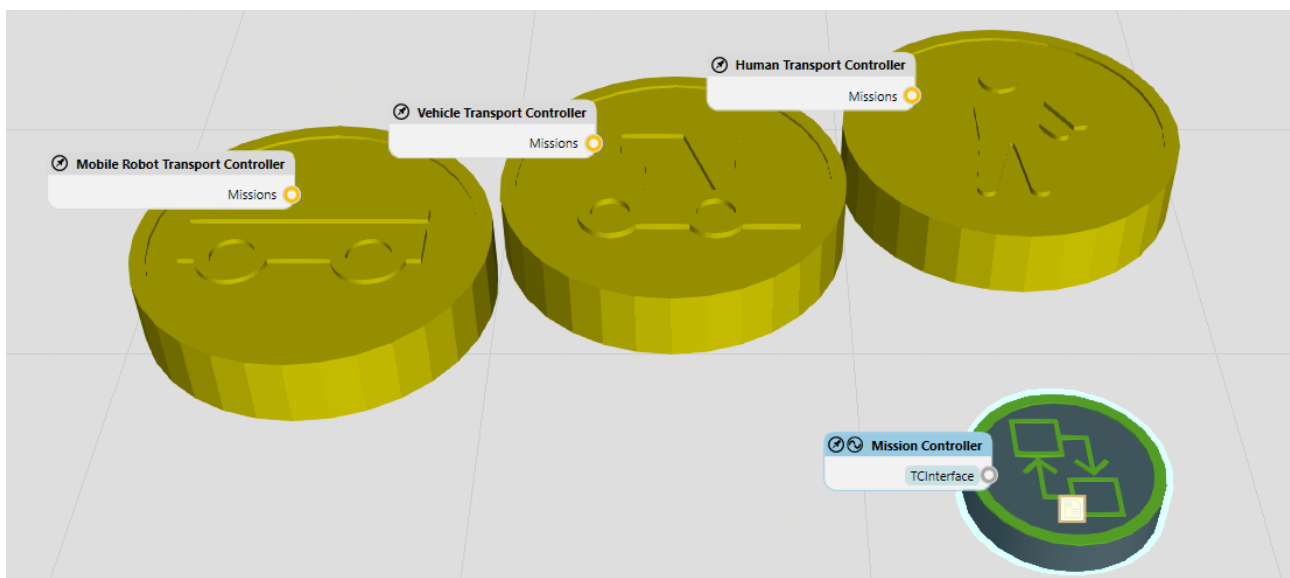
Missions allows you:

- Define a proper task flow for resource perspective.
- Define *optional* steps which can be skipped if transport or work task is not available.
- Execute *custom* actions, i.e., other than PM transport and work tasks.
- Define via points (Move).
- Define a battery level requirement for a route (Charge).
- Synchronize the resource with other resources and processes (SendSignal, WaitSignal)
- Simulate a “robot mobile platform” (RunRobotRoutine)
- Call “submissions” (CallMission)

Compatibility

The Process Modeling (PM) mission feature is supported by the PM components within the eCatalog of **Visual Components 4.7**, or later. **The main component is the Mission Controller**, which is compatible with the following transport controllers:

- Mobile Robot Transport Controller
- Human Transport Controller
- Vehicle Transport Controller



Limitations

PM Missions are based on Process Modeling framework, which means that you need to define the products, processes, and flow in the Process tab. If you are not familiar with Process Modeling, see the related tutorials in <https://academy.visualcomponents.com/>.

Also note that:

- TransportIn and TransportOut tasks must be matched. The resource cannot collect a product if the product does not have transportation solution.
- The transport task must be associated with the same controller that the mission controller is connected to.
- Similarly, the work task must be called by a Work statement in the process for the same controller that the mission controller is connected to.
- Missions are not dynamic; they must be pre-defined and cannot change after sending them to the transport controller.
- Mission execution must take time. If a mission instance is executed without spending time, the whole mission is cancelled. This can happen if the collect steps are optional, but there is no product available. You can avoid this by adding at least two Move steps with different locations in the mission to make the resource moving in a loop.
- Load and Unload Assist by using another resource is not supported with the missions in the version Visual Components 4.7.

How to use

To use Missions, follow the following steps:

1. First, build up your PM layout with the latest eCatalog 4.7 components. If you are using an existing layout, please make sure that they are correct versions, and thus, supporting this feature. If needed, replace the components with the latest versions.
2. Add mission controller to the layout. You can find it in the eCatalog 4.7, in PM Navigation collection.
3. Connect the mission controller with “Interfaces” tool to a transport controller, or in the Component Properties panel, press the button "Connect All Mission Controllers”.

Note: When the mission controller is connected, the “normal” functionality of the transport controller is disabled. Thus, if there are no missions defined, no transportation or work tasks are handled. It is not possible to mix the missions and “freely distributed” tasks in the same transport controller.

4. Select the controller, and in the component properties panel, go to Missions tab and define a new mission. You need to enable the mission and define its activator. Then you define the mission steps. See the next section for more information.
5. User interface in the Mission tab is used to populate the content in the Note, which defines the actual data for missions. You can edit the data also in the JSON-format if the syntax is correct. Modifying the note, updates the properties in the User interface accordingly.
6. Run the simulation. If the mission has been defined correctly, you should see a message in the Output window like the following:

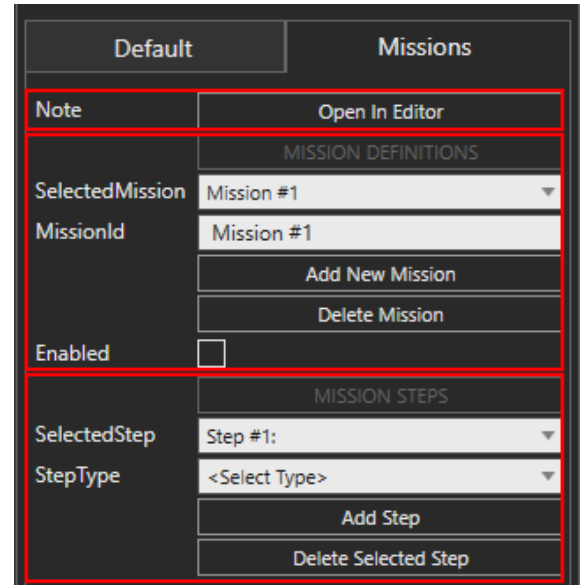
(0.0) [Human Transport Controller - INFO]: ScheduledMission (id = "Mission #1" with unlimited occurrences) defined.

7. If you face problems, see the debugging section.

Defining missions

There are three sections in the Mission Controller user interface:

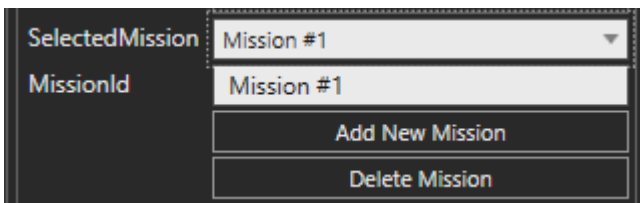
- **Note** allows you to:
 - open, view and modify the definitions in editor in JSON format.
- **Mission Definitions** allows you to:
 - Select, rename, add, or remove missions.
 - Enable/disable the selected mission.
 - Define the selected mission activators.
- **Mission Steps** allows you to:
 - Select, add, or remove step.
 - Change selected step type.
 - Define the other step properties.



Mission Definitions

The mission definition specifies what is the mission Id (unique name) and allows you to define its activators.

Adding, Removing and Renaming Missions



SelectedMission. Defines which mission you are modifying. Its value field has a drop-down menu of all the missions that are currently defined.

MissionId. To rename the selected mission change the value of this field.

Add New Mission. By default, there is one mission called Mission #1. If you want to add a new mission, click the *Add New Mission*-button. This will create a new mission, add it to the step properties of *SelectedMission* and select it.

Duplicate Mission. Pressing this button creates a new mission with a new id and copies all the properties to it from the selected mission, including the activators and mission steps.

Delete Mission. Removes the selected mission.

Activators

Enabled allows you to turn on or off the mission. If the mission is not enabled, its steps can be still executed by using `CallMission-step` from another mission.

Priority defines the priority of the mission. If there are multiple missions that can be activated at the same time, the missions with higher priority are instantiated first. In some occasions, the lower priority mission instance might need to wait for a resource to be free. However, the running mission instances are always prioritized according to their start time.

The screenshot shows a configuration panel for a mission activator. The settings are as follows:

- Enabled:**
- Priority:** 10
- Activator:** Scheduled
- StartTime:** 50 s
- Recurring:**
- RecureEvery:** 80 s
- LimitOccurenc...:**
- Occurrences:** 100
- AcceptedReso...:** Selected Resources
- SelectedResou...:** Human (Anna)
- PickResource:** Human (Anna)
- Buttons:** Add Resource, Remove Resource

Activator. If enabled, activators define when mission is instantiated during simulation run, how many instances there could be in total and how often they occur.

You can select among the following activators:

- **Scheduled.*** The first instance is scheduled to start at *StartTime* and after that, a new instance could be started at every time interval defined by *RecurringInterval*.
- **Repeated.**** The first instance is scheduled to start at *StartTime* and after the mission instance is completed, the same instance could be recycled to start over after delay defined *DelayBetween*.
- **Triggered.*** When the defined Boolean signal is triggered to True, a new mission instance is started.

NOTES:

* **Instantiating more missions than can be handled by the resources can cause excess usage of memory (overhead) over time.**

** **It is ok to set *DelayBetween* value to zero. In this case, the mission is repeated immediately after completing the previous instance.**

If a mission instance is executed without spending time, the mission is cancelled. This can happen, e.g., if the collect steps are optional, but there is no product available. You can avoid this by adding at least two Move steps with different locations in the mission to make the resource move in a loop.

StartTime defines when the first occurrence of the mission is started.

Recurring, RecurringInterval. If *recurring*, a new occurrence is triggered after every time interval of *RecurringInterval*.

DelayBetween. When *Activator* is set to *Repeated*, a new occurrence is started after the time defined by *DelayBetween* has elapsed from the moment the previous occurrence was completed.

LimitOccurrences, Occurrences. If *LimitOccurrences* is *False*, the mission is starting new occurrences all the time that the simulation runs. Else, mission is disabled, after the occurrence index defined with *Occurrences* is reached.

SignalComponent, SignalName. Defines what is the name of the signal that starts a new occurrence of the mission, and which component has it.

Accepted Resources

AcceptedResources. You can also specify which resources the mission instances can be allocated to:

- **Any** does not limit the accepted resources. Any of the connected resource can be used.
- **SelectedResources** allows you to pick the selected resources. Make sure that the mission controller and the resources are connected to the transport controller, to see the resources in the *PickResource* list.

SelectedResources contains a list of the resources that can be used for the mission.

PickResource allows you to select a connected resource from the list. If the drop-down menu is empty, connect the mission controller and the resources to the transport controller first.

Add Resource adds the resource selected in the *PickResource* to the list of *SelectedResources*.

Remove Resource removes the resource selected in the *PickResource* from the list of *SelectedResources*.

Mission Types

The mission steps define what is the workflow within the mission; what actions to do and in which order. The mission steps can be executed by using *CallMission*-step from another mission, even if the mission is not enabled.

By default, there is already an empty step. Define first its properties. Then create another empty step with the *Add Step*-button, then define its properties.

SelectedStep. Defines which step you are modifying. Its value field has a drop-down menu of all the steps that are currently defined.

StepType defines what action is executed in this step. The following types are supported:

- **CallMission** replaces the step with all the steps from the mission defined with the *MissionToCall* property. The activators of the called mission is not affecting, and it does not matter if the called mission is enabled or not.
- **Charge** step checks the current capacity of the resource and goes charging in one of the *ChargingLocations* if the limit is under the percentage defined by *ToChargeLimit* of its full

capacity. When *ToChargeLimit* is reached, the resource stops charging and continues executing the mission.

NOTE: The resource must have PowerManager enabled, thus the human resource cannot execute this step.

- **Collect** step picks a product from the specific *TransportNode* or from any process defined by *ProcessGroupId*. If multiple products are available, and the resource has capacity, all products can be picked if *CollectAll* is set to True. If there are no products available, the resource will wait for a product, unless the *OptionalStep* is set to True.

NOTES:

- **TransportIn and TransportOut tasks must be matched. The resource cannot collect a product, if the product does not have transportation solution.**
 - **The mission must have a deliver step after the collect step, and the transport node of the collect must match with the source and the transport node of the deliver step must match with the destination of a transportation solution.**
 - **The transport task must be associated with the same controller that the mission controller is connected to.**
- **Deliver** step drops a product to the specific *TransportNode* or to any process defined by *ProcessGroupId*. If multiple products are available, all products can be dropped if *DeliverAll* is set to True.
 - **Move** step navigates the resource to a position defined by *Component* location, the component's *PositionFrame*, or the world position defined by the *Position* vector. If *AlignToTarget* is True, the resource rotates to align with the target or *Theta* value.

NOTES:

- **The resource is using the PM Navigation System to move to the target. That means, it will utilize the connected PathWays and avoid obstacles whenever possible.**
 - **The resource moves to location where the component or frame location was when defined. If the target is moving, the target location will not update.**
- **MoveJoint** step moves the joint defined by *JointName* of *Component* to *TargetValue*. If value of *Component* is **Null**, the resource moves its own joint with the defined *JointName*. The *MotionTime* specifies the motion time in seconds and can be set to **0 s** to let the controller plan the motion.
 - **RunRobotRoutine** step calls the *Robot* to call the routine defined by *RoutineName*. The resource will wait the robot to complete the routine if *WaitExecution* is True, else it will continue the mission immediately.
 - **SendSignal** step sends a *BooleanSignal* of *Component* with *SignalValue*.

- **Wait** step makes the resource to delays the execution of the mission for *WaitTime* in seconds.
- **WaitSignal** waits that the value of *BooleanSignal* of *Component* is *SignalValue*. If *WaitTrigger* is True, it does not consider the current value of signal when the step is started. *TimeOut* defines maximum amount of simulation time in seconds to wait and can be set to zero to have no time limit.
- **Work** step executes an active work task in the specific *TransportNode* or in any process defined by *ProcessGroupId*. If there is no active work task available, the resource will wait for one, unless the *OptionalStep* is set to True.

Add Step. Adds an empty step in the end of the list. If you need to modify the order of the steps, use the editor to change the JSON text. Be careful with the syntax - remember to leave the last comma out at the end of the list.

Delete Selected Step removes the step that is currently in the value of the *SelectedStep* property.

Move Up changes the order of the steps by moving the selected step one step earlier.

Move Down changes the order of the steps by moving the selected step one step later.

Note Editor

The note editor can be opened in the icon in 3D viewport on top of a mission controller, or from its Component Properties-panel under Missions-tab.

The content of the note is automatically populated from the component properties in the Missions-tab of its owner component but works also other way around - modifying the note triggers updating the component properties.

The note is in JSON format, and the syntax must be correct – otherwise the component properties are not updated, and the missions aren't correctly working during simulation. That is why it is recommended to use the component properties. However, it is handy to copy a step or change the order of the steps with a cut-and-paste or copy the whole mission definitions from component to another.

Note: Be careful with the syntax - remember to leave the last comma out at the end of the list.

Troubleshooting

When encountering any problem while using PM Missions, please double check the following:

- Make sure that the components are compatible with each other. If you are unsure, or using an old layout, replace the transport controllers, resources, and mission controllers with the latest versions from the eCatalog.
- Make sure the mission controller and the resources are connected to a transport controller.
- See first the Output panel for any messages. If the mission has been defined correctly, you should see an info message in the Output window when running the simulation, such as the following:
(0.0) [Human Transport Controller - INFO]: ScheduledMission (id = "Mission #1" with unlimited occurrences) defined.
- If you have modified the content of *Missions::Definitions* -note, make sure that the format is correct.
- Make sure mission is Enabled.
- Make sure mission's activator returns True.
 - In case of Scheduled or Repeated missions, the simulation must be run until the start time is elapsed to see the first mission instance activated.
 - In case of Triggered missions, the activator signal must be triggered to activate a mission.
- If you are using Collect, Deliver or Work step as the first step, the resource might be waiting for the task to arrive from the controller, and it might look like the resource is not doing anything. In that case, it is recommended to use Move step as the first step.
- Check the value of CurrentState property of a resource.
- There are some debug tools for Visual Components Professional and Premium users, including hidden DEBUG-property.
- See the [Visual Components Forum](#) for more information and updates.