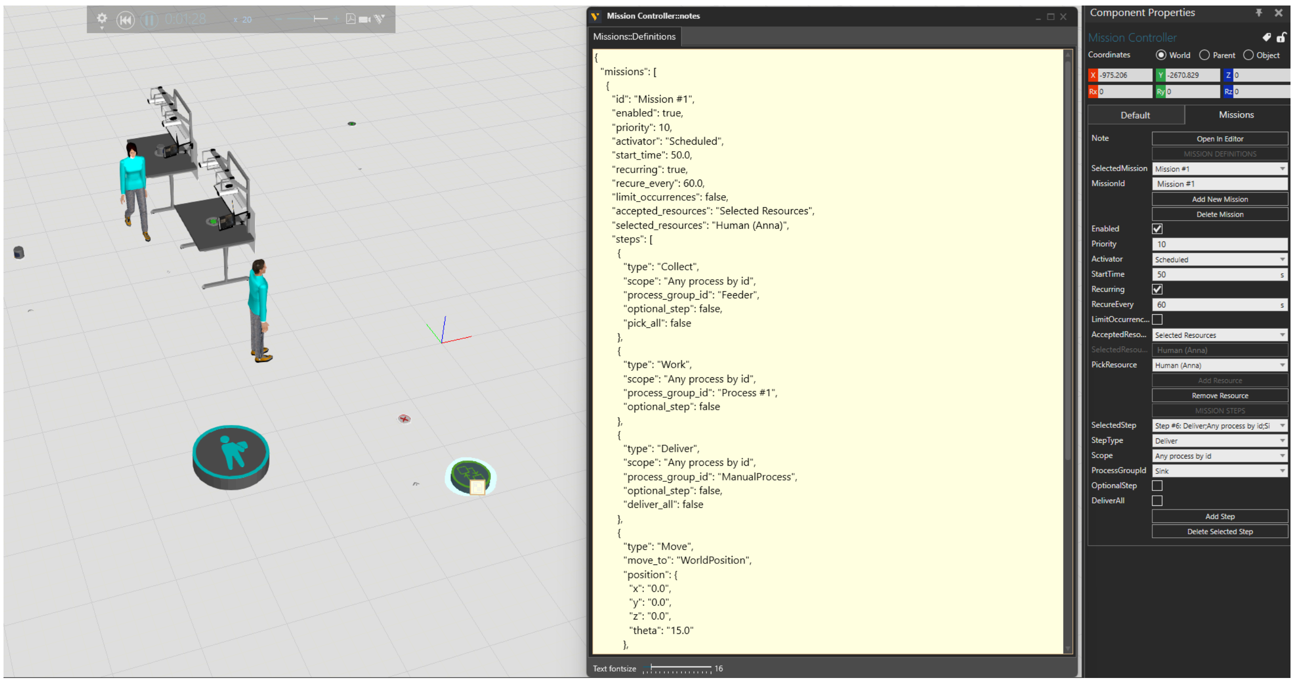
Process Modeling Missions – Manual

Visual Components 4.7 | Version: June 25th, 2023



```
{
  "missions": [
    {
      "id": "Mission #1",
      "enabled": true,
      "priority": 10,
      "activator": "Scheduled",
      "start_time": 50.0,
      "recurring": true,
      "recure_every": 60.0,
      "limit_occurrences": false,
      "accepted_resources": "Selected Resources",
      "selected_resources": "Human (Anna)",
      "steps": [
        {
          "type": "Collect",
          "scope": "Any process by id",
          "process_group_id": "Feeder",
          "optional_step": false,
          "pick_all": false
        },
        {
          "type": "Work",
          "scope": "Any process by id",
          "process_group_id": "Process #1",
          "optional_step": false
        },
        {
          "type": "Deliver",
          "scope": "Any process by id",
          "process_group_id": "ManualProcess",
          "optional_step": false,
          "deliver_all": false
        },
        {
          "type": "Move",
          "move_to": "WorldPosition",
          "position": {
            "x": "0.0",
            "y": "0.0",
            "z": "0.0",
            "theta": "15.0"
          },
```

Coordinates: (● World) (Parent) (Object)

X -975.206	Y -2670.829	Z 0
Rx 0	Ry 0	Rz 0

SelectedMission	Mission #1
MissionId	Mission #1
Enabled	✓
Priority	10
Activator	Scheduled
StartTime	50 s
Recurring	✓
RecureEvery	60 s
LimitOccurrenc...	☐
AcceptedReso...	Selected Resources
SelectedResou...	Human (Anna)
PickResource	Human (Anna)
SelectedStep	Step #6: Deliver;Any process by id;Si
StepType	Deliver
Scope	Any process by id
ProcessGroupId	Sink
OptionalStep	☐
DeliverAll	☐

Missions is a feature of the Process Modeling (PM) library that allows you to define a 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

Mission.	Refers typically to a Mission instance but can also mean mission definition in some contexts.
Mission controller.	User interface to define mission definitions and place a holder to contain them. This data is sent to the connected transport controller on simulation start in JSON format.
Mission definition.	Definition of a mission from which mission instances can be created in simulation. Contains all the information needed to create the mission instances of this type.
Mission instance.	One occurrence of the mission created during simulation and defined by the mission definition.
Process Modeling (PM).	Process Modeling is a feature provided by 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.
Sub-mission	Set of steps defined by another mission definition that are called from the “parent” mission or another sub-mission. Any mission, except the parent mission can be called regardless of whether the mission is enabled or not.

Capabilities

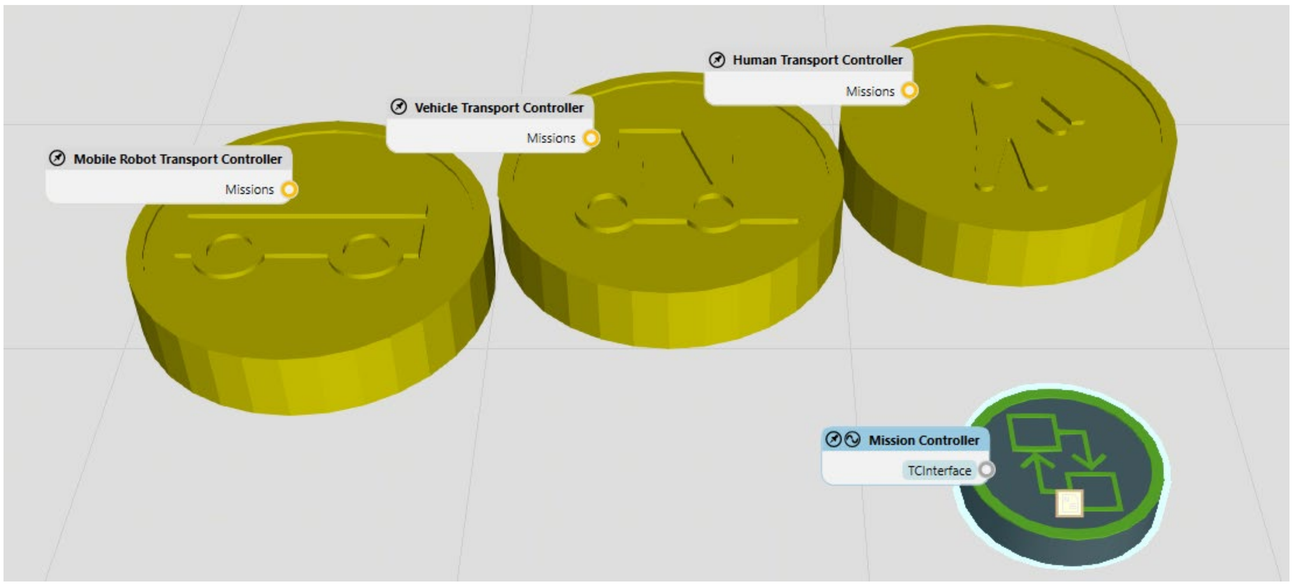
Missions allow you to:

- Define a proper task flow for resource perspective.
- Define *optional* steps which can be skipped if a 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 sub-missions (CallMission).

Compatibility

The Process Modeling (PM) mission feature is supported by the PM components within the eCatalog of Visual Components 4.7 or higher. **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


Human Transport Controller
Missions
Vehicle Transport Controller
Missions
Mobile Robot Transport Controller
Missions
Mission Controller
TCInterface

Limitations

PM Missions are based on the Process Modeling framework, meaning that you need to define the products, processes, and flow in the Process tab. If you are not familiar with Process Modeling, refer to the related tutorials in the academy at 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 a 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 be changed after they have been sent 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 occur 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, so that the resource moves in a loop.
- Load and Unload Assist by using another resource is not supported by missions in Visual Components 4.7.

How to use

To use Missions, follow the following steps:

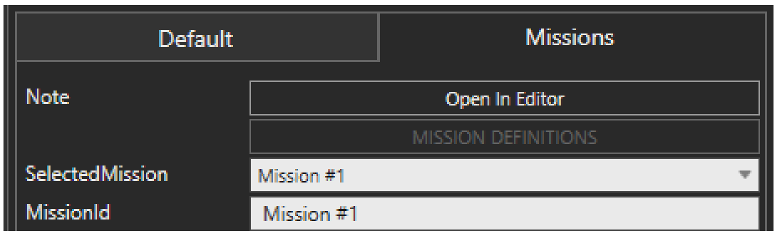
1. First, build up your PM layout with the latest Visual Components 4.7 eCatalog components. If you are using an existing layout, make sure that it includes the latest component versions that support this feature. If needed, replace components with the latest versions.
2. Add a mission controller to your layout. You can find it in the eCatalog in the PM Navigation collection.
3. From the **HOME** tab, connect the mission controller with the **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 the **Missions** tab and define a new mission. You need to enable the mission and define its activator. Then you define the mission steps. Refer to the next section for more information.
5. The user interface of the **Missions** tab is used to populate the content in the **Note**, which defines the actual data for missions. You can also edit the data in JSON-format if the syntax is correct. Modifying the **Note** updates the properties in the user interface accordingly.
6. Then run the simulation. If the mission has been defined correctly, you should see a message like the following in the **Output** window:

   (0.0) [Human Transport Controller - INFO]: ScheduledMission (id = "Mission #1" with unlimited occurrences) defined.
7. If you experience any issues, refer to the Troubleshooting section.

Defining missions

There are three sections in the Mission Controller user interface:



Note

Allows you to open, view and modify the definitions in JSON format in the editor

Mission Definitions

The *MISSION DEFINITIONS* specifies the mission Id (unique name) and allows you to define its activators. You can do the following:

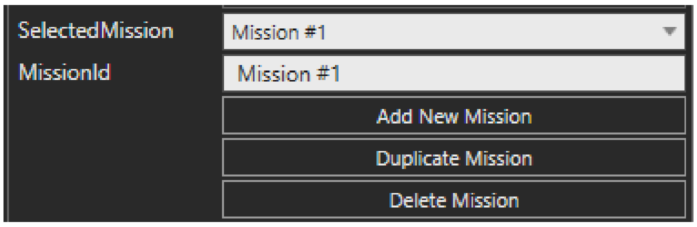
- Select, rename, add, or remove missions.
- Enable or disable the selected mission.
- Define the selected mission activators.

SelectedMission

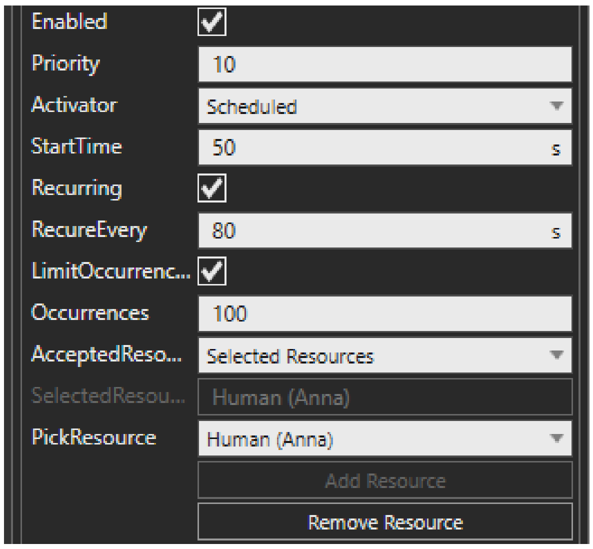
Defines which mission you are modifying. Its value field includes a drop-down menu of all the missions that are currently defined.

MissionId

To rename the selected mission edit 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** copies all properties of the selected mission and creates a new mission with a new id including activators and mission steps.
- **Delete Mission** Removes the selected mission.



Enabled	✓
Priority	10
Activator	Scheduled
StartTime	50 s
Recurring	✓
RecureEvery	80 s
LimitOccurrenc...	✓
Occurrences	100
AcceptedReso...	Selected Resources
SelectedResou...	Human (Anna)
PickResource	Human (Anna)

Enabled

Allows you to turn on or off the mission. If the mission is not enabled, its steps can still be executed by using **StepType** *CallMission* 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 instances, a lower priority mission might need to wait for a resource to be free. However, the running mission instances are always prioritized according to their start time.

Activator

If enabled, activators define when a mission is instantiated during a simulation run, how many instances there could be in total and how often they occur. You can select among the following activators:

Scheduled

- **StartTime** defines when the first occurrence of the mission is started.
- **Recurring** When True a new occurrence is triggered after every time interval of **RecurringInterval**.
  - If **LimitOccurences** is False, the mission is starting new occurrences every time the simulation runs.
  - Or when True after the occurrence index defined with **Occurrences** is reached the mission is disabled.

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

Repeated

- The first instance is scheduled to start at **StartTime** and after the mission instance is completed, the same instance could be repeated starting after a delay defined by **DelayBetween**.
- If **LimitOccurences** is False, the mission is starting new occurrences every time the simulation runs.
  - Or when True after the occurrence index defined with **Occurrences** is reached the mission is disabled.

NOTE: It is ok to set the **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.

Triggered

- When the defined Boolean signal is triggered to True, a new mission instance is started. **NOTE:** Other activators first create a mission instance in defined simulation time based on **StartTime** value. **Triggered** does not create instances based on simulation time, but when signal is triggered.
- **SignalComponent** and **SignalName**. Defines the name of the signal that starts a new occurrence of the mission, and which component has it.
- If **LimitOccurences** is False, the mission is starting new occurrences every time the simulation runs.
    - Or when True after the occurrence index defined with **Occurrences** is reached the mission is disabled.

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

Filtering 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 resources can be used.
- *SelectedResources* allows you to pick the selected resources. To see the resources in the **PickResource** list, make sure that the mission controller and the resources are connected to the transport controller.
- **SelectedResources** contains a list of the resources that can be used for the mission.
- **PickResource** contains a list of available resources when the mission controller and the resources are connected to the transport controller.
- **Add Resource** adds the resource selected in the **PickResource** list to the list of **SelectedResources**.
- **Remove Resource** removes the resource selected in the PickResource from the list of SelectedResources.

Mission Steps

The *MISSION STEPS* allow you to:

- Select, add, or remove a step.
- Change selected step type.
- Define the other step properties.

Mission Types

The mission steps define the workflow within the mission; what actions to perform 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 and 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 are not affected and it does not matter whether the called mission is enabled or not.
- *Charge* step checks the current capacity of the resource and goes to charge 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.

  NOTE:

  - *TransportIn* and *TransportOut* tasks must be matched. The resource cannot collect a product if the product does not have a transportation solution.
  - The mission must have
    - A deliver step after the collect step.
    - The transport node of the collect step must match with the source of the transport task (the transport node of the feeding process).
    - The transport node of the deliver step must match with the destination of the transport task (the transport node of a needing process).
  - 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.

  NOTE:

  - 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 a 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 the value of **Component** is *Null*, the resource moves its own joint with the defined **JointName**. **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 for 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 will make the resource delay the execution of the mission for **WaitTime** in seconds.
- *WaitSignal* waits that the value of **BooleanSignal** of **Component** with **SignalValue**. If **WaitTrigger** is True, it does not consider the current value of the signal when the step is started. **TimeOut** defines maximum amount of simulation time in seconds to wait and can be set to zero so it has 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 to the end of the list. If you need to modify the order of the steps, use the editor to change the JSON syntax. However be careful with the syntax - remember to leave the last comma out at the end of the list.

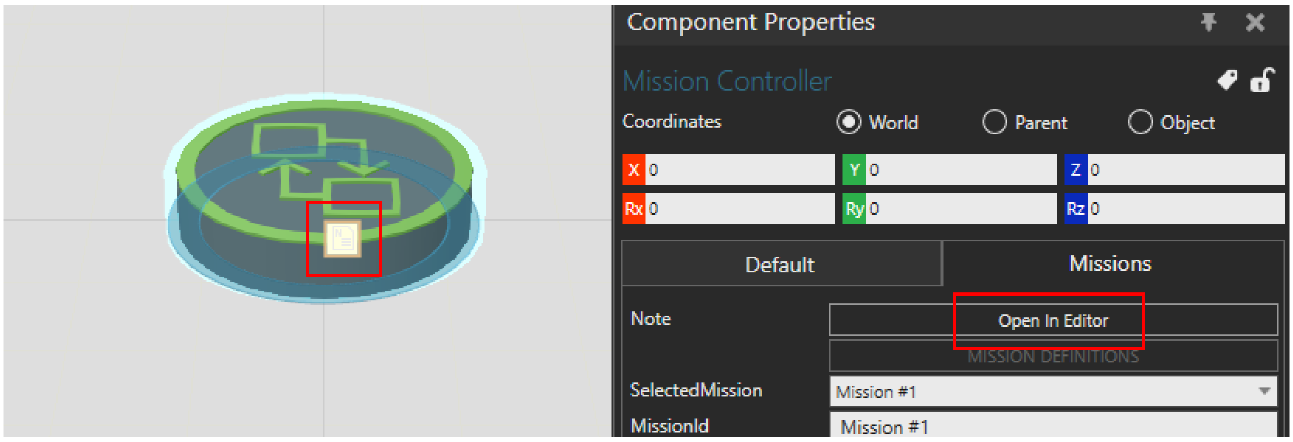
Delete Selected Step removes the step that is currently in the value of the *SelectedStep* property.

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 using the icon on top of a mission controller in 3D viewport, or from its **Component Properties** panel under the **Missions** tab.



Coordinates: [✓ World] Parent Object

X 0 Y 0 Z 0

Rx 0 Ry 0 Rz 0

Note

SelectedMission: Mission #1

MissionId: Mission #1

The content of the note is automatically populated from the **Component Properties** in the **Missions** tab of its owner component, but also works the 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 will not function correctly during simulation. That is why it is recommended to use the **Component Properties**.

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 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.
- First check the **Output** panel for any messages. If the mission has been defined correctly, you should see an info message 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 the **Missions::Definitions** note, make sure that the format is correct.
- Make sure mission is Enabled.
- Make sure the 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* steps 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.