

Connectivity configurator addon

Quick guide

Paired Variables		
Component	Simulation variable	ServerVariable
Mover	ActPos	Axes.Mover[1].ActPos
CRB 1100-4/0.580	IRCS_Axis1_Axis1	Robot.Axes.Axis 1
CRB 1100-4/0.580	IRCS_Axis2_Axis2	Robot.Axes.Axis 2
CRB 1100-4/0.580	IRCS_Axis3_Axis3	Robot.Axes.Axis 3
CRB 1100-4/0.580	IRCS_Axis4_Axis4	Robot.Axes.Axis 4
CRB 1100-4/0.580	IRCS_Axis5_Axis5	Robot.Axes.Axis 5
CRB 1100-4/0.580	IRCS_Axis6_Axis6	Robot.Axes.Axis 6
XPlanar Mover #2	Controller_A_A	Axes.PlanarMover2[A].ActPos
XPlanar Mover #2	Controller_B_B	Axes.PlanarMover2[B].ActPos

Introduction

This document gives an overview of the Connectivity configurator addon. Aim of the addon is to make it faster and easier to pair multiple variables in the Visual Components connectivity plugin connections. This is achieved by defining the paired variables in a spreadsheet that can include templates and be easily modified.

The addon initially supports only Beckhoff ADS variable pairing.

How to use

Building the variable pairing spreadsheet

Open the template, that is provided together with this document, in the spreadsheet tool of choice. The template includes the first row with pre-defined headers, but the same format can be created manually on to an empty sheet.

	A	B	C	D	E	F	G	H	I
1	Plugin type	Address	Port	Direction	Simulation component name	Simulation Behaviour Name	Simulation Property Name	ServerTagName	
2	ADS		851	server to simulation	Mover #{i}		ActPos	Axes.Mover[{i}].ActPos	
3	ADS		851	server to simulation	CRB 1100-4/0.580	IRC5	Axis(i).VALUE	Robot.Axes.Axis {i}	
4	ADS		851	server to simulation	XPlanar Mover {i}	Controller	{a}	Axes.PlanarMover(i)[{a}].ActPos	
5	ADS		851	simulation to server	Destaco S01{a} Parametric	ServoController	Joint_{i}	GVL.S01{a} Parametric {i}	

The first row includes headers for each column in the following order:

- Plugin type – Currently only *ADS*.
- Address – The IP address of the server. Can be left blank for localhost.
- Port – The server port. Default *851* for TwinCAT3 PLC.
- Direction – Data flow direction. Defines if the variable pair is sending or receiving data from the server. Set to either *server to simulation* or *simulation to server*.
- Simulation component name – The simulation component name for the paired variable. You can use templates to find multiple similar components and match all suitable variables with one row.
- Simulation behaviour name – Name of the simulation component behaviour that is used for variable pairing. Blank for component properties.
- Simulation property name – Name of the simulation property that should be paired with the given server tag name. Can be a component property, or a behaviour property. Write the port index for signal map behaviours, and leave blank for other signal type behaviours.
- Server tag name – Name of the server tag / variable. If your simulation variable includes a template, make sure to use a matching template also for the server tag name. Dot notation is used to define the paired variable from the server structure, for example: *GVL.bStart*.

One row can be used to create multiple variable pairs by using templates {i} or {a}.

- {i} is used to represent an integer number.
 - Example 1: Match variables from multiple components with a similar name. *MyComponent #{i}* looks for all components that start with *MyComponent* and attempts to find the matching variable from the server.
 - Sometimes the starting index may vary between 0 and 1, so you can use {i+1} or {i-1} to create variable pairs.
 - Example 2: Pairing *my_sim_variable[0]* with *my_server_variable[1]*, etc. is done as follows: *my_sim_variable[{i-1}] - my_server_variable[{i}]*.
- {a} is used to represent an alphabet.
 - This is useful when the variables use changing letter instead of a number.
 - Example: the template *joint.{a}* would match *joint.X*, *joint.Y*, *joint.Z* from the corresponding component.

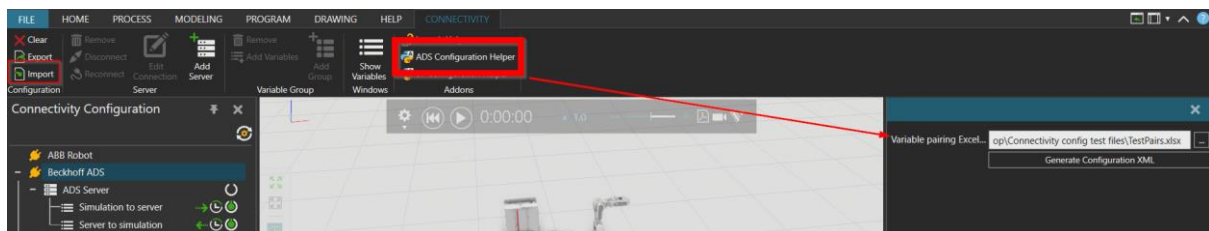
Once you have defined all desired variable pairs you can save the spreadsheet with the .x/sx file extension. Finally, close the file so that it can be read by the addon.

Visual Components

The addon is installed by downloading the .zip package and extracting the content to the *My Commands* folder typically found in: *C:\Users\%USERNAME%\Documents\Visual Components\VERSION\My Commands*

Restart the Visual Components application, and load a simulation layout that you want to use connectivity plugins with. Note that this addon requires you to have a specific layout open that has components that match the ones defined in the spreadsheet. Otherwise the result will be an empty configuration.

The addon should appear in the *Connectivity* tab under *Addons* group. Click on the Configuration Helper button and select the spreadsheet containing variable pairing definitions. Note that the file cannot be open in other applications simultaneously.



Click on the Generate Configuration button, and observe the Output panel at the bottom of the screen for additional information. If successful, the Output panel will list the variables that were found from the simulation, and where the configuration file is saved. The default location is set to the addon installation folder inside My Commands.

Click on the Configuration *Import* button on the left side of the top toolbar to import the configuration created by the addon. The default file name is ServerConfiguration.xml. New plugin connections should be created for the set plugin type. Ensure that the correct variables are paired, and edit the variable group properties as needed.

Finally, establish the connection and start the simulation and server to test that the variables are synchronized properly.