

Auto-Connect Port Prototypes

Technical Reference

Version 1.0

Authors	Pavol Gramblicka
Status	Released

Document Information

History

Author	Date	Version	Remarks
Pavol Gramblicka	2015-04-28	0.1.0	Initial version
Pavol Gramblicka	2015-04-28	1.0.0	Final version

Contents

1	Overview	5
1.1	Intended Audience	5
1.2	Terms and Acronyms	5
2	Input Port Prototypes	6
3	Auto-Connect.....	7
3.1	Auto-Connect – simple pattern.....	7
3.2	Auto-Connect – enhanced patterns.....	9
3.2.1	Prefix, Postfix.....	9
3.2.2	Complex patterns using the component prototype name.....	10
4	Contact.....	13

Illustrations

Figure 2-1	Start the contextless auto-connecting	6
Figure 2-2	Start the context dependent auto-connecting.....	6
Figure 3-1	Simple auto-connect scenario.....	7
Figure 3-2	List of proposed connections	8
Figure 3-3	Context dependent auto-connecting of a prefixed port prototype	9
Figure 3-4	Context dependent auto-connecting with a prefixed naming pattern	10
Figure 3-5	Context dependent auto-connecting of a port prototype without the component restriction	11
Figure 3-6	Context dependent auto-connecting of port prototypes with the component restriction	12

Tables

Table 1-1	Terms and Acronyms	5
-----------	--------------------------	---

1 Overview

DaVinci Developer is part of Vector's solution for AUTOSAR compatible software design. It is used to design and configure software components and provides various concepts to support this process.

This document describes a part of the design process related to DaVinci Developer from a technical point of view, trying to give the user a better understanding of the internal processes and how the tool reacts in different situations.

1.1 Intended Audience

This document aims at developers who are involved in the AUTOSAR design process and use DaVinci Developer to integrate various software components.

1.2 Terms and Acronyms

Term	Definition
DEV	DaVinci Developer

Table 1-1 Terms and Acronyms

2 Input Port Prototypes

The auto-connect functionality may be started contextless from the **Software Design** toolbar (Figure 2-1) or from the context menu of the graphic view (Figure 2-2).



Figure 2-1 Start the contextless auto-connecting

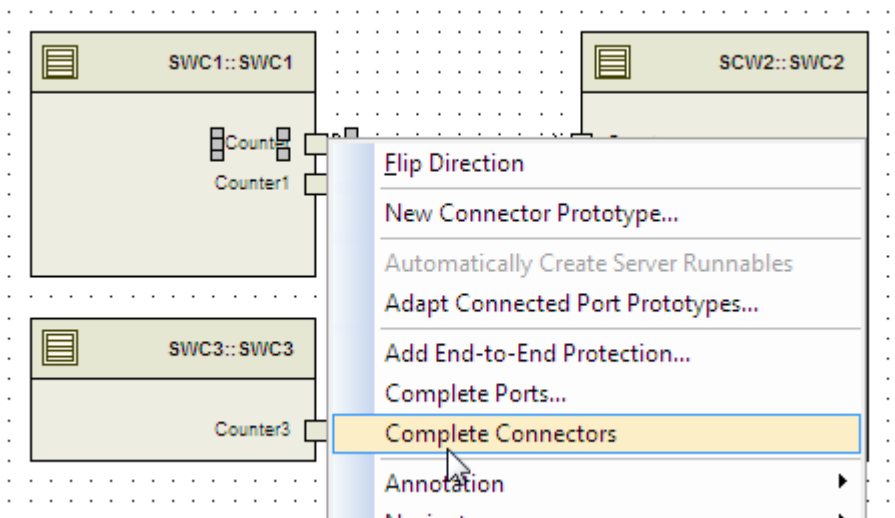


Figure 2-2 Start the context dependent auto-connecting

1. In case of the contextless connecting, all port prototypes of the parent composition are considered.
2. In case of the context dependent connecting, the connections are created according to the provided selection.
 - > **no selection** – all port prototypes within the composition are considered
 - > **a component is selected** – all port prototypes of the component are considered
 - > **single port prototype selection** – only these port prototypes are considered to be the source resp. destination of a proposed connector prototype

3 Auto-Connect

3.1 Auto-Connect – simple pattern

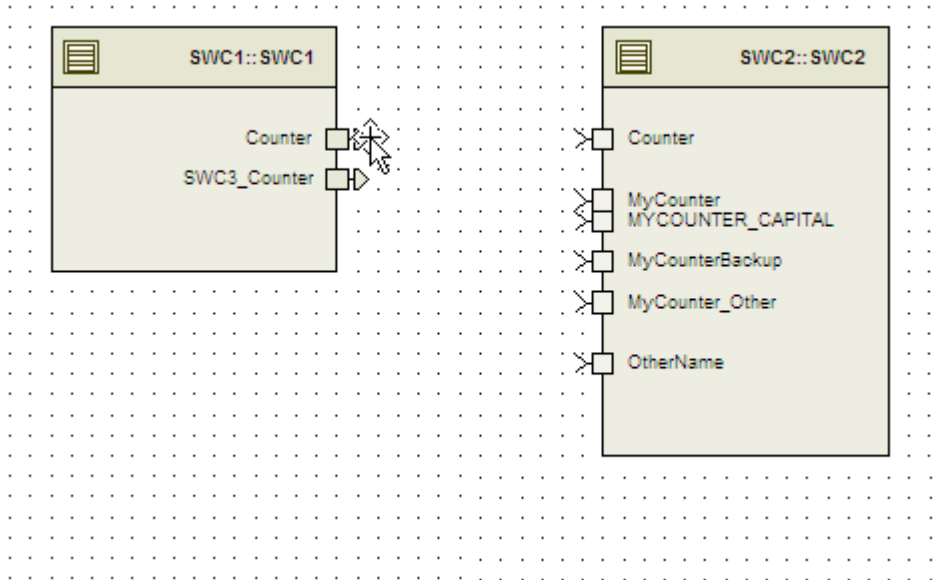


Figure 3-1 Simple auto-connect scenario

Imagine we want to use the context dependent auto-connect for the port prototype **Counter** of the component **SWC1**.

1. Select the port prototype
2. Right mouse-click on port prototype to open the context menu
3. Choose **Complete Connectors**

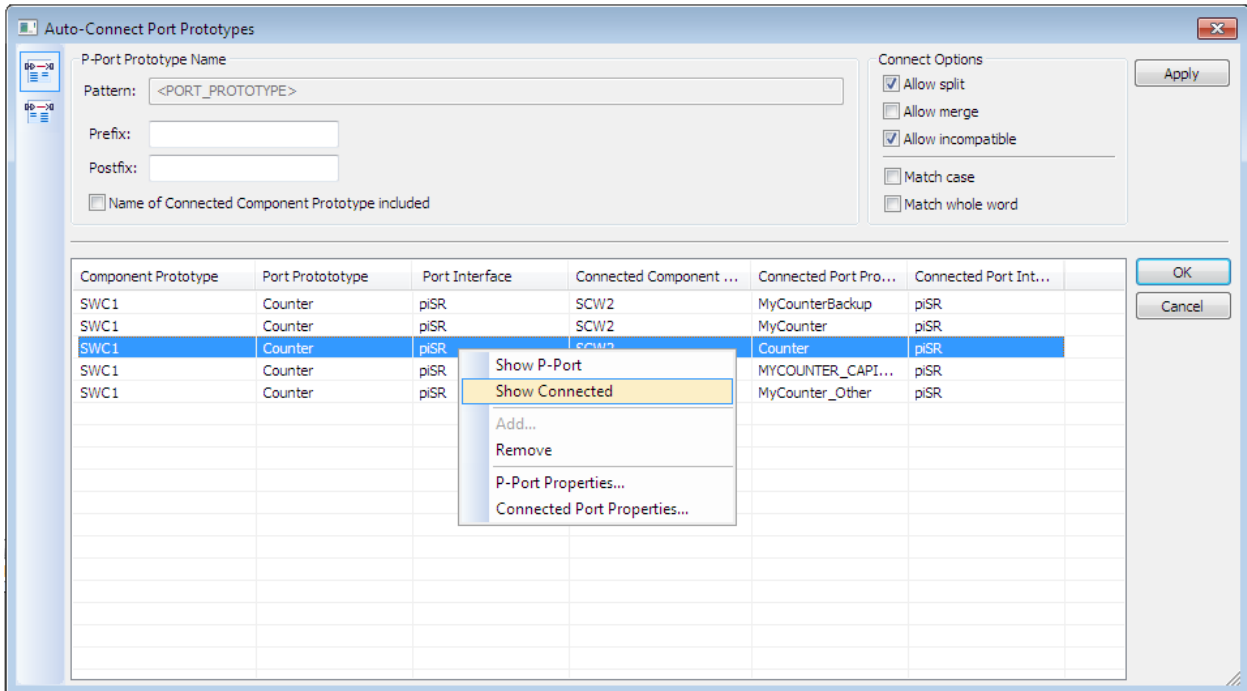


Figure 3-2 List of proposed connections

The algorithm is looking for counterparts for the selected port prototype **SWC1.Counter**

- > in the first step, the naming pattern is analyzed to provide a key string
- > simple pattern without prefix resp. postfix is used i.e. only the name of the port prototype is essential for the search algorithm, key string = **Counter**
- > for all connectable port prototypesⁱ, if the key string is a substring of the name of a connectable port prototype a connection will be proposed
- > a line in the dialog then corresponds to the proposed connection
- > additionally the algorithm can be refined with **Connect Options**
 - > **Match case** – consider case sensitive matching
 - > **Match whole word** – consider exact matching
 - > **Allow split** – in case of a P-Port more than one connection is allowed
 - > **Allow merge** – in case of an R-Port more than one connection is allowed
 - > **Allow incompatible** – consider AUTOSAR compatibility rules
- > If you refined the options, press **Apply** to take the changes into account
- > to get more information about proposed connection press right mouse button to open the context menu
 - > with **Show P-Port** resp. **Show Connected** menu buttons the port prototypes might be located in the software design
 - > with **Add** and **Remove** menu buttons the proposed list might be edited
 - > with the rest menu buttons the property dialogs might be open

3.2 Auto-Connect – enhanced patterns

In a more complicated case, the naming pattern of the selected port prototype has to be identified.

3.2.1 Prefix, Postfix

Consider the designer uses always a prefix or a postfix for his port prototypes, e.g. **SWC1::PPCounter**. In this case, the algorithm cannot find any connectable port prototype which includes the substring **PPCounter** in his name.

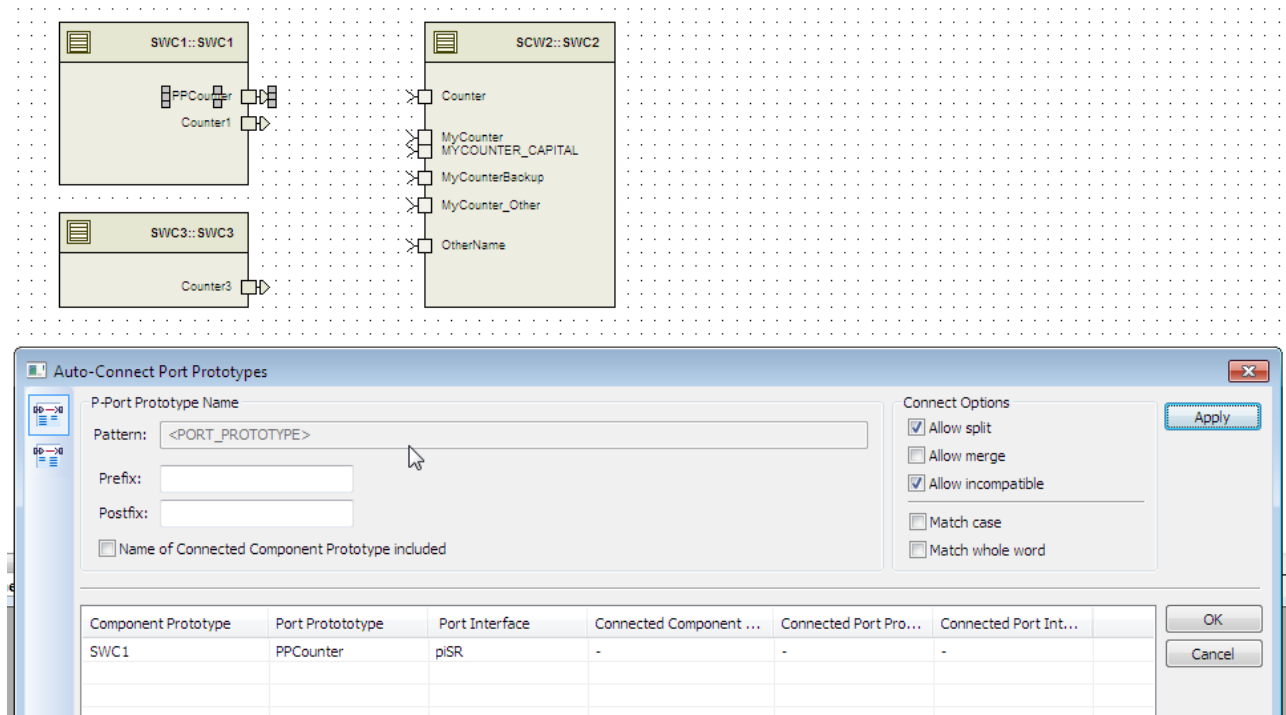


Figure 3-3 Context dependent auto-connecting of a prefixed port prototype

The user must explicitly determinate the prefix part. When computing the key string, the prefix part will be removed and the algorithm is running like in the simple case.

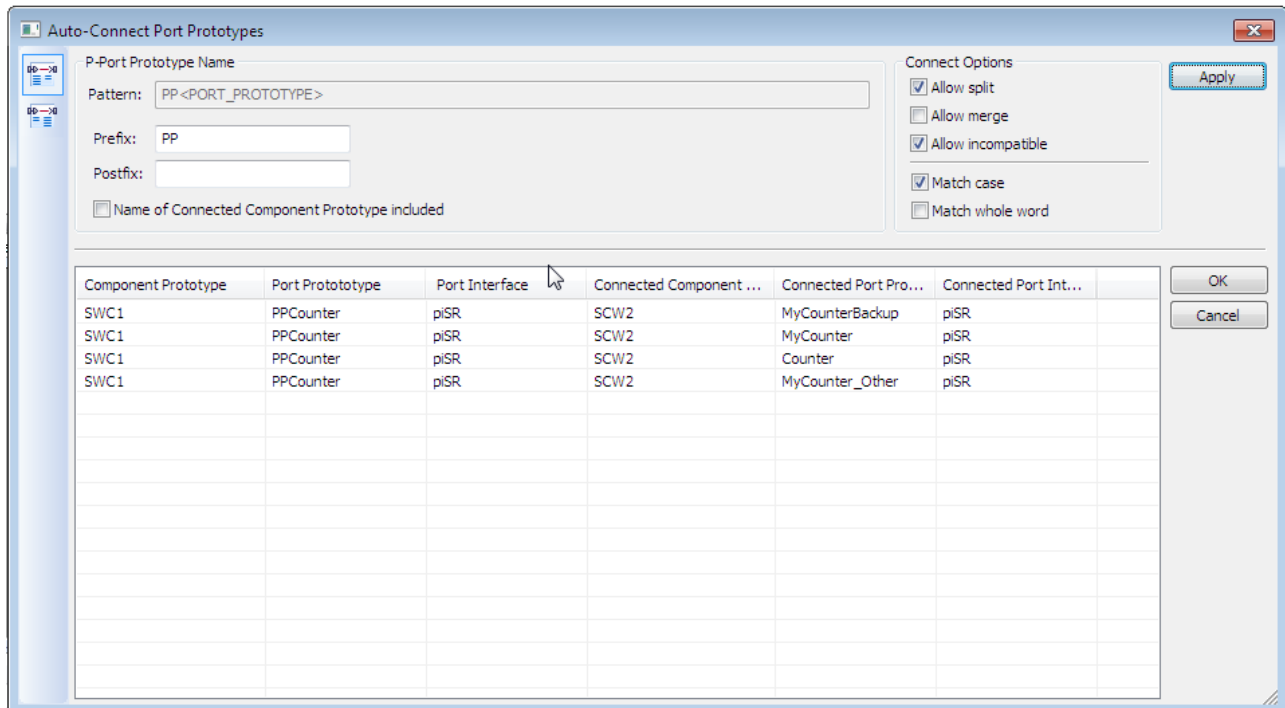


Figure 3-4 Context dependent auto-connecting with a prefixed naming pattern

3.2.2 Complex patterns using the component prototype name

In a very special case, the destination component may be identified by the naming pattern. This scenario is applicable if there are components which have port prototypes of the same or similar naming but the proposed set should be restricted to a dedicated component. Suppose we want to restrict the proposal to component **SWC2**. The Figure 3-5 displays such a scenario where connections to **SWC2** and **SWC3** will be proposed. In our example, we can simply remove the connection to **SWC3** but in general it might be difficult to identify such relations.

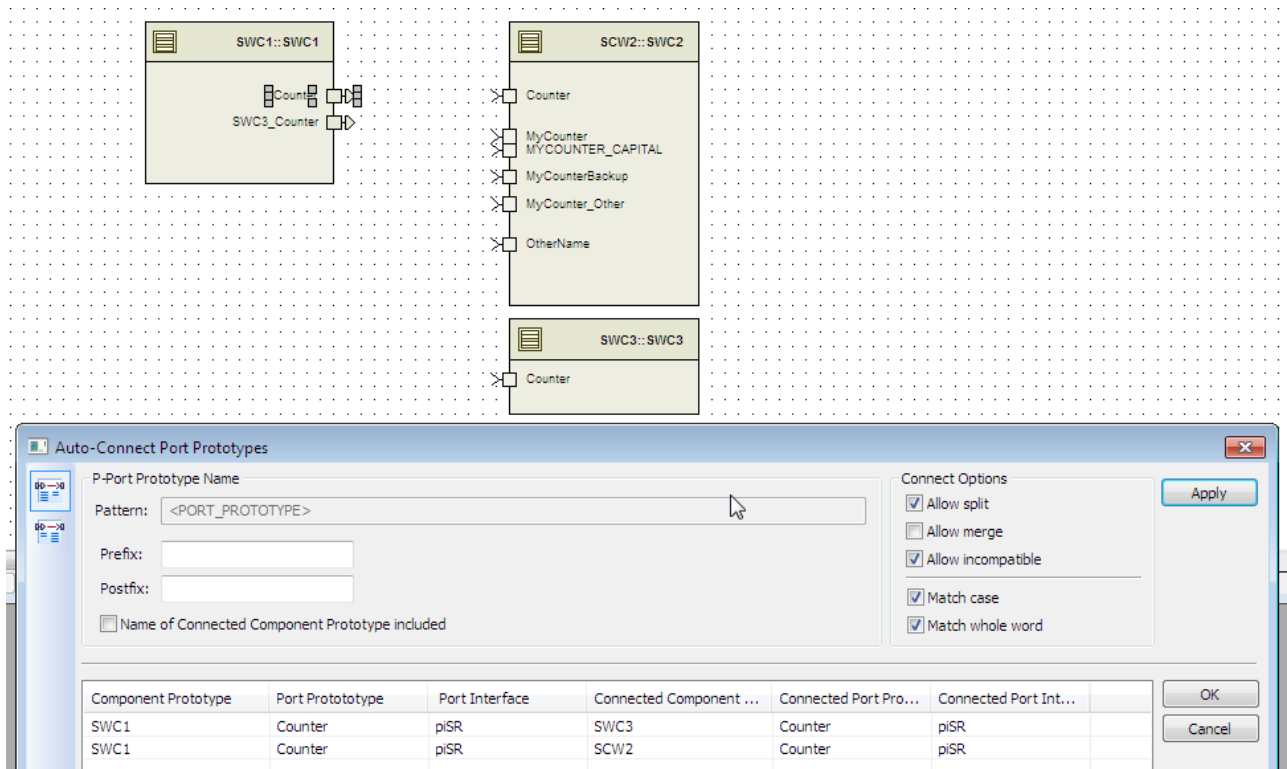


Figure 3-5 Context dependent auto-connecting of a port prototype without the component restriction

To be sure the port prototype of the desired component will be proposed, the restriction must be coded in the name of the selected port prototyped. The naming pattern looks like this <COMPONENT_PROTOTYPE>_<PORT_PROTOTYPE>. The algorithm will parse the port prototype name and try to identify corresponding parts. Additionally to the current concept, the name of the component will be matching.

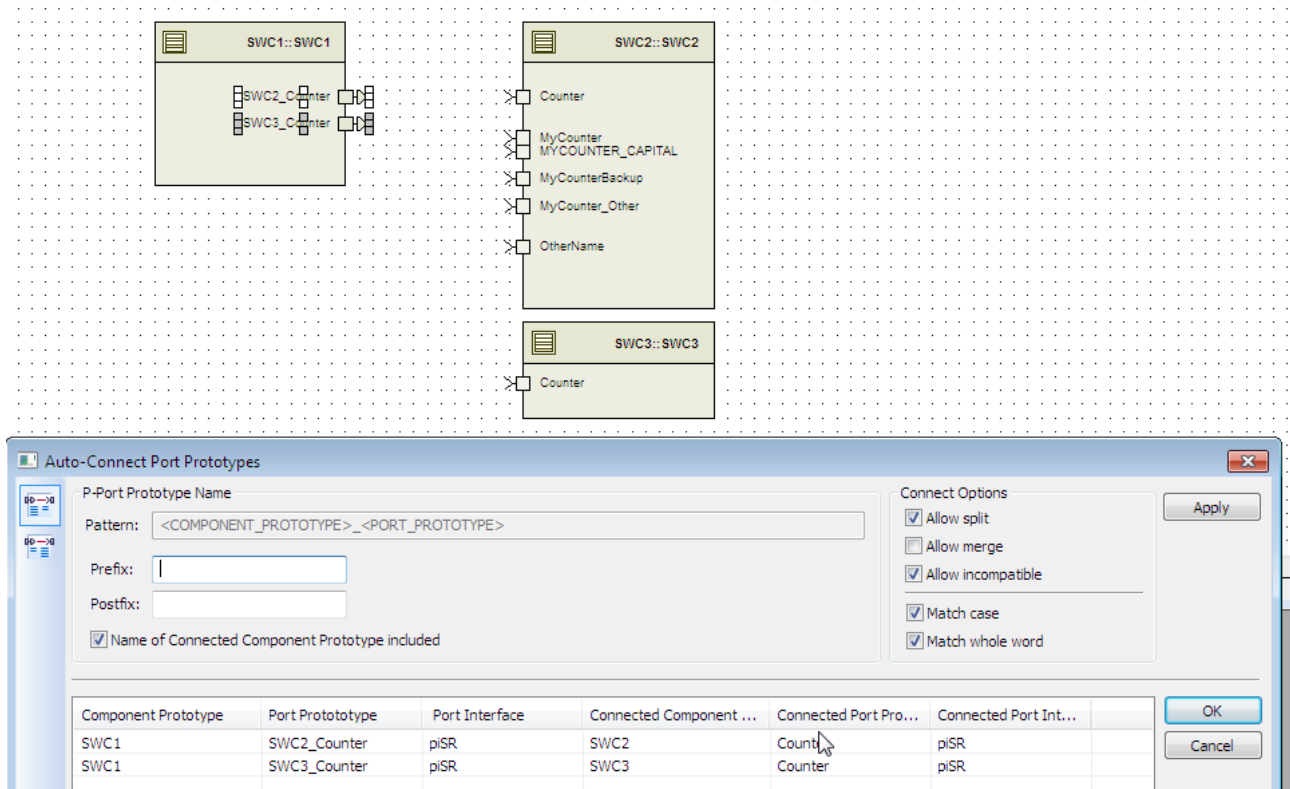


Figure 3-6 Context dependent auto-connecting of port prototypes with the component restriction

4 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

www.vector.com

ⁱA connectable port prototype is such a port prototype which allow for creation of a meaningful connector prototypes. In case of an assembly connector, if P-Port is selected only an R-Port or PR-Ports are meaningful and vice versa. In case of a delegation connector prototypes, if P-Port is selected only a P-Port or PR-Port are meaningful and similar if an R-Port is selected only an R-Port or PR-Port are meaningful.