



# <AutomationML/>

**The Glue for Seamless  
Automation Engineering**

**Application Recommendation Extension:  
Extension Rack for Automation Project  
Configuration**

**Document Identifier: ARE APC ER, V 1.2.0**

**State: September 2020**

©AutomationML consortium

Version 1.2.0, September 2020

Contact: [www.automationml.org](http://www.automationml.org)

**Table of contents**

Table of contents .....	3
List of tables .....	4
1 Introduction.....	5
1.1 Basics.....	5
1.2 References.....	6
2 General notes regarding exchange of Automation Project Configuration data .....	7
3 Modelling of Extension Racks in Automation Project Configuration data with AutomationML .....	8
3.1 Subnet.....	8
3.2 Node.....	8
3.3 CommunicationInterface .....	8
3.4 CommunicationPort .....	8
Appendix A AutomationML Examples .....	9
A.1 Extension Rack .....	9

## List of tables

Table 1 – Overview of AutomationML parts.....	5
---	---

## 1 Introduction

A very frequently occurring task within the planning process of production and automation systems is the exchange of automation project configuration information of automation system devices between ECAD and PLC systems. To avoid multiple engineering in the participating systems ECAD and PLC systems need an interface for sharing this information.

In case of beginning engineering in the ECAD tool certain rules must be observed to get the hardware information in the correct location in the PLC tool. In case of beginning engineering in the PLC tool non placed functions must be placed and operated in the ECAD tool.

The application recommendation “Automation Project Configuration” describes these workflows and the method of hardware configuration modelling using AutomationML and refers in several items to bus types. Because not all existing and upcoming bus types can be described in the application recommendation “Automation Project Configuration” the specific extensions of each bus type shall be described in a separate bus specific specification.

This application recommendation extension describes the specific extensions for Extension Racks.

### 1.1 Basics

The data exchange format AutomationML which is standardising in the IEC 62714 standard is a neutral, free, and XML-based data format. It has been developed in order to support the data exchange between engineering tools in a heterogeneous engineering tool landscape.

Due to the different aspects of AutomationML the IEC 62714 consists of different parts.

*Table 1 – Overview of AutomationML parts*

Part / Document Identifier	Title	Description
Part 1 / WP Arch, V 2.0.0	Architecture and general requirements	This part specifies the general AutomationML architecture, the modelling of the engineering data, classes, instances, relations, references, hierarchies, basic AutomationML libraries and extended AutomationML concepts.
Part 2 / WP Lib V 2.0.0	Role class libraries	This part specifies additional AutomationML libraries.
Whitepaper / WP Comm V 1.0.0	Communication	This Whitepaper describes the modelling of Communication mechanisms in AutomationML
Whitepaper / WP eClass V 1.0.0	AutomationML and eCl@ss integration	This Whitepaper describes the integration of eCl@ss in AutomationML
Best Practice Recommendation / BPR MlingExp V 1.0.0	Multilingual expressions in AutomationML	This Whitepaper describes the handling of different texts for different languages in AutomationML
Best Practice Recommendation / BPR RefDes V 1.0.0	Modelling of Reference Designations	This Whitepaper describes the handling of reference designations following IEC 81346-1:2009-07 within AutomationML

## 1.2 References

The following documents are referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

*Extensible Markup Language (XML) 1.0:2004, W3C Recommendation (available at <<http://www.w3.org/TR/2004/REC-xml-20040204/>>)*

*IEC 62424:2008, Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

*Whitepaper AutomationML Part 1 – AutomationML Architecture, November 2018*

*Whitepaper AutomationML Part 2 – AutomationML Role Libraries, October 2014*

*Whitepaper AutomationML – AutomationML Communication, September 2014*

*Whitepaper AutomationML – AutomationML and eCI@ss Integration, November 2017*

*Best Practice Recommendation Multilingual expressions in AutomationML, March 2017*

*Best Practice Recommendation Modelling of Reference Designations, September 2017*

*Application Recommendation Automation Project Configuration, April 2020*

## **2 General notes regarding exchange of Automation Project Configuration data**

The AutomationML export of Automation Project Configuration data is based on the application recommendation "Automation Project Configuration". This recommendation only describes extensions, additional definitions or not needed objects from AR APC related to the communication technology of Extension Racks. Extensions are realized by derivation from abstract base class extension defined by AR APC.

### 3 Modelling of Extension Racks in Automation Project Configuration data with AutomationML

#### 3.1 Subnet

A “**Subnet**” is not supported for this bus type.

#### 3.2 Node

A “**Node**” is not supported for this bus type

#### 3.3 CommunicationInterface

A “**CommunicationInterface**” is defined in AR APC.

The attribute “Type” shall have the value “ExtensionRack”.

Per DeviceItem only one ExtensionRack interface of type “LogicalEndpoint” with multiple Ports shall be used.

#### 3.4 CommunicationPort

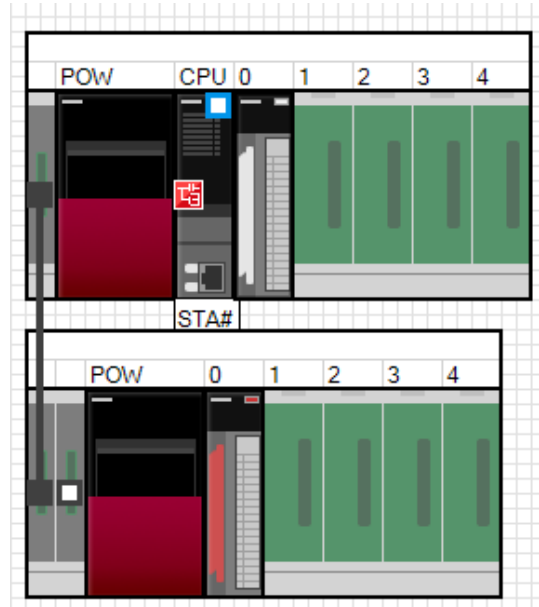
A “**CommunicationPort**” is defined in AR APC.



## Appendix A AutomationML Examples

### A.1 Extension Rack

In the following it is shown how a rack extension can be modelled. The rack “R35B” is extended with the rack R65B.



Both racks contain an InternalElement with the Role “CommunicationInterface” with the attribute “Type” set to “ExtensionRack”. The “CommunicationInterface” InternalElement consists of “CommunicationPort” InternalElements, which contain “CommunicationPortInterface” Interfaces. The “CommunicationPortInterfaces” are linked to each other to model the rack extension.

