SIEMENS

SIMATIC

PROFINET
PROFINET Driver for Controller Engineering Interface

Programming and Operating Manual

11/2015

A5E32593517-AB

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E32593517-AB
℗ 10/2015 Subject to change

Preface

Preface

Purpose of the manual

This user documentation describes the XML data format of the engineering interface of the PROFINET Driver for Controller.

Target group for the manual

This manual is intended for software and application developers who want to configure PROFINET IO controllers without using a Siemens engineering system (TIA Portal, STEP 7, NCM). Knowledge of the PROFINET IO standard is a prerequisite.

Developers receive a CD with the complete source code of the PROFINET Driver for Controller, the documentation, IO controller application examples with appropriate XML configurations and a sample platform porting for Microsoft Windows and Linux.

Structure of the manual

This manual describes the XML data format of the engineering interface of the PROFINET Driver for Controller. It is structured as follows:

- Section 1: Introduction
- Section 2: XML structure
- Section 3: PNIO classes, attributes and links
- Section 4: Used data types
- Section 5: PNIO data blocks
- Section 6: Description of PNIO data blocks
- Appendix: Abbreviations / Glossary of terms

This manual includes the description of the XML data format of the engineering interface of the PROFINET Driver for Controller at the time of release. We reserve the right to update the user documentation in light of new product releases.

Guide

The manual contains various navigation aids that allow you to find specific information more quickly:

- A complete table of contents as well as a list of all figures and tables are provided at the beginning of the manual.
- In the appendix you will find a list of abbreviations and a glossary, which define important technical terms used in this manual.

Conventions

The terms "PROFINET Driver for Controller" and "PN Driver" are used synonymously in this manual.

Please observe notes labeled as follows:

Note

A note contains important information about the described product, about handling the product or about a specific section of the documentation that requires special consideration.

Additional user documentation

Besides this manual there are additional manuals for the PROFINET Driver for Controller:

- PROFINET IO-Base user programming interface manual
- How to Port PN Driver V1.1 parameter manual
- Quick Start PN Driver V1.1 getting started manual
- Online application example for PROFINET Driver on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109478514)

Additional support

If you have questions regarding the described PROFINET Driver for Controller that are not addressed in the documentation, please contact your local representative at the Siemens office nearest you.

Please send questions, comments and suggestions regarding this manual in writing to the specified e-mail address.

In addition, you will find general information, current product information, FAQs and downloads that can be useful on the Internet (http://www.siemens.com/comdec).

Technical contact for Germany/worldwide

Siemens AG
ComDeC (http://www.siemens.com/comdec)

Phone: +49 911 750 2080
Phone: +49 911 750 4384
Phone: +49 911 750 2078
Fax: +49 911 750 2100
E-mail: (mailto:ComDeC@siemens.com)

Office address:
Würzburger Str. 121
90766 Fürth, Germany

Postal address:
P.O. Box 2355
90713 Fürth, Germany

Technical contact for the USA

PROFI Interface Center
(http://www.profiinterfacecenter.com)
One Internet Plaza
Johnson City, TN 37604

Phone: +1 (423) 262-2576
Fax: +1 (678) 297-7289
E-mail: (mailto:PIC.industry@siemens.com)

Technical contact for China

PROFI Interface Center China
7, Wangjing Zhonghuan Nanlu
100102 Beijing

Fax: +86-10-6476 4725
E-mail: (mailto:Profinet.cn@siemens.com)

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. You can find more information about industrial security on the Internet (http://www.siemens.com/industrialsecurity).

To stay informed about product updates as they occur, sign up for a product-specific newsletter. You can find more information on the Internet (http://support.automation.siemens.com).

Table of contents

Introduction 1

PROFINET is an automation concept for implementing modular, distributed applications. PROFINET allows you to create automation solutions, which are familiar to you from PROFIBUS. The engineering tool for the PROFINET Driver for Controller may be Siemens TIA Portal or a non-Siemens tool that is able to use the open engineering interface described in this manual.

In TIA Portal, PROFINET Driver for Controller can be used without a license. In other words, you may install TIA Portal V13 SP1 and the PN Driver HSP on your PC and continue using TIA Portal to create PN Driver projects as long as you want, even after the trial license period of TIA Portal has expired. Please note that in this case, you can only use devices in your projects which can also be used without a license, e. g. GSDML based IO devices.

A good knowledge of PROFINET IO is required to use the PROFINET Driver for Controller.

The consistency of the configuration must be checked by the engineering system. The engineering interface of the PROFINET Driver for Controller assumes that the information and structure of engineering data are valid.

Engineering data are passed to the PNIO controller in XML format in a single file. This XML file contains the information required for configuring the PROFINET IO controller.

The content of the file will be passed to the function SERV_CP_Startup() as byte array. For SERV_CP_startup(), refer to manual "IO-Base user programming interface", file "(...)\doc\manual\PGH_IO-Base_76.pdf" on the CD.

Content and target audience of this interface description

This document is intended for developers of PROFINET IO controllers. It contains:

- Overview of the XML structure of the engineering interface of the PROFINET Driver for Controller
- Description of the contained PNIO classes, attributes and links
- Description of the used data types and contained PNIO data blocks

This documentation does **not** include:

- Overview of PROFINET
- Description of the PROFINET protocol
- Detailed description of the PROFINET IO stack structure and processes

1.1 Object model of the user interface

Object model based on class/object concept

The user interface of the PN Driver automation system follows an object model. This object model describes the IO controller's view upon an IO system (IO devices, their submodules and data records) used by this IO controller.

The object model is based on the class/object concept, which means an object is instantiated from a class.

Each object has an object type, which means it is instantiated from an object class. The object class may hold the information which is type-specific, for example, the type-specific attribute values of an object.

1.2 Identification

Identification of classes, attributes and links

Classes, attributes and links between objects are identified by identifiers.

Note

Names of objects and attributes in the XML file are **not** used for identification - they are only used to make the XML file easier to read. PN Driver only uses the identifiers (numbers).

Classes: ClassRID (Class Runtime Identifier)

The ClassRID identifies a class.

Attributes: AID (Attribute Identifier)

The AID identifies an attribute of an object. The attribute of an object can be a **variable** or a **key**; see section XML elements (Page 12).

Links: AID (Attribute Identifier)

Objects can be connected to other objects by links. The links are identified by an AID identifier. Currently, PN Driver uses only one type of link – it connects the IO controller object to the corresponding distributed IO system object. For details on links, see section Relationship between IOcontroller and distributedIOsystem (Page 14).

ClassRID numbers, AID numbers and the AID number for links are listed in section PNIO classes, attributes and links (Page 22).

1.3 Module and ModuleProxy

As specified in PROFINET IO, modules do not implement productive functionality, they are only meant to be an aggregating container for the (productive) submodules.

Modules are indexed by their slot number. Modules always have submodules. Submodules are the actual source of IO data.

ModuleProxy

A ModuleProxy is used to hold the diagnostics address of a module when there is more than one submodule within this module. However, a module may consist of only one submodule - in this case this submodule represents also the module for diagnostics and no ModuleProxy is necessary.

General rule as to whether or not a ModuleProxy is necessary:

- If a module has more than one submodule, it additionally contains a ModuleProxy submodule which serves as module representative for module diagnostics. The ModuleProxy always gets the special subslot number 0xFFF0.
- If a module carries exactly one submodule, this submodule fulfills two tasks simultaneously: being the module representative for module diagnostics and being the submodule itself.

With PROFINET IO a module never has parameters. All parameters are modeled at the submodule level.

Module

Within a module, there are the following typical use cases:

- Module with a single submodule having no IO data. In this case the module aggregates a submodule (e. g. ET 200S power module). In this use case, there is no explicit ModuleProxy submodule to represent the module; this submodule fulfills both tasks simultaneously.
- Module with a single submodule having IO data. In this case the module aggregates an IO submodule (e. g. ET 200S DI16). In this use case, there is no explicit ModuleProxy submodule to represent the module; this submodule fulfills both tasks simultaneously.
- Module with multiple submodules with or without IO data. In this case the module aggregates:
  - a submodule representing the module; the so-called “ModuleProxy” used for module diagnostics - ClassRID 8
  - an IO submodule for each submodule having IO data - ClassRID 10
  - a submodule for each submodule not having IO data - ClassRID 9

1.4 Logical base address

Diagnostics address for hardware objects

The **LADDR** (logical base address) is used as diagnostics address for hardware objects like interfaces, ports, modules and submodules. This address is used in the context of alarms/diagnostics and reading/writing of records. PROFINET IO specifies addressing of an hardware object (e. g. reading a record from a submodule) via its geographic address (interface number, station number, slot number and subslot number). Since the geographic address is impractical to handle, the logical address LADDR is also available.

XML structure 2

2.1 XML elements

Used XML elements

The XML interface of PN Driver uses the following XML elements:

Table 2- 1 XML elements

XML element	Description
Object	**Object** contains an object of a specific class. The class type is identified by the child element **ClassRID**.
ClassRID	**ClassRID** contains a class identifier.
Key	**Key** is an object attribute. It is used for indexing the station number, slot number, subslot number and data records.
Variable	**Variable** is an object attribute that represents the actual data of an object. The type of the variable is defined by its child element **AID**. Each variable has an XML attribute **Name** which is used to label the XML element **Variable**.
AID	**AID** contains the attribute identifier of the variable. It defines the type of the variable.
Value	**Value** contains the value of the variable. It can contain further child elements (**Field** or **Element**).
Field	**Field** contains PROFINET IO record data. Field has two XML attributes: **Key** and **Length**. **Key** contains the record index, **Length** contains the length of the record.
Link	**Link** contains a link to another object. It contains two child elements, the **AID** of the link and the **RID** of the target object (**TargetRID**).
TargetRID	**TargetRID** is a child element of the XML element **Link** and contains the **RID** of the target distributedIOsystem. It is generated by the engineering system and is unique within the scope of the XML file.
RID	**RID** contains a unique ID of a distributedIOsystem. This identifier is generated by the engineering system and is unique within the scope of the XML file.
Element	**Element** contains IOmapping. It is used to hold the input addresses, output addresses, and their lengths, of an IO submodule.

Example of IOmapping

```
<Variable Name="IOmapping">
  <AID>5</AID>
  <Value Datatype="Scalar" Valuetype="STRUCT">
    <!— Ibase: input address is 3 -->
    <Element AID="6" Datatype="Scalar" Valuetype="UINT32">3</Element>
    <!— Ilength: length of input is 1 byte -->
    <Element AID="7" Datatype="Scalar" Valuetype="UINT16">1</Element>
    <!— Qbase: no output address since Qlength is 0 -->
    <Element AID="8" Datatype="Scalar" Valuetype="UINT32">0</Element>
    <!— Qlength: length of output is 0, meaning no output address -->
    <Element AID="9" Datatype="Scalar" Valuetype="UINT16">0</Element>
   </Value>
</Variable>
```

2.2 Structure of the XML file

Overview of XML structure

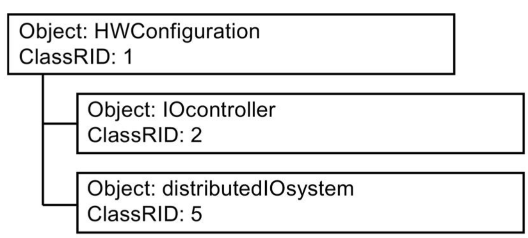

Figure 2-1 XML file structure

Structuring

```
<Object Name="HWConfiguration">
     <ClassRID>1</ClassRID>
     <Object Name="PN Driver_1">
          <ClassRID>2</ClassRID>
           ...............
     </Object>
     <Object Name="PROFINET IO system">
          <ClassRID>5</ClassRID>
           ...............
     </Object>
</Object>
```

Principle

The structure of the XML file is very simple - it basically consists of objects and their attributes (keys and variables). Objects and attributes are defined with unique IDs. The structure is self-explanatory, therefore no XML schema is provided. Due to its simplicity the XML structure can be easily extended for future features and no versioning is required. PNIO parameters are packed into data records and only the engineering system and PN Driver need to know the structure of these data records.

2.3 Relationship between IOcontroller and distributedIOsystem

Main parts of the XML file

The XML document basically consists of two main parts:

- IOcontroller
- distributedIOsystem

These two parts are connected by a “link”, which connects a PROFINET IOInterface of an IOcontroller with a distributedIOsystem.

A link has an AID (which is the type of the link) and a TargetRID which is the key to the actual linked object (in this case the distributedIOsystem).

The RID (runtime identifier) is generated by the engineering system. This RID must be unique for all links within an XML file. Currently, PN Driver only supports one PROFINET interface so there is only one link and only one RID used.

Note

You may use a constant value like "12345" for the RID.

Example of a link

```
<Object Name="HWConfiguration">
     <ClassRID>1</ClassRID>
     <Object Name="PN Driver_1">
         <ClassRID>2</ClassRID>
         <Object Name="PROFINET interface_1">
              <ClassRID>3</ClassRID>
              <Link>
                   <AID>16</AID>
                   <TargetRID>12345</TargetRID>
              </Link>
         </Object>
     </Object>
     <Object Name="PROFINET IO system">
        <RID>12345</RID>
        <ClassRID>5</ClassRID>
        ...............
     </Object>
</Object>
```

2.4 IOcontroller

Overview of IOcontroller structure

The IOcontroller is structured as follows:

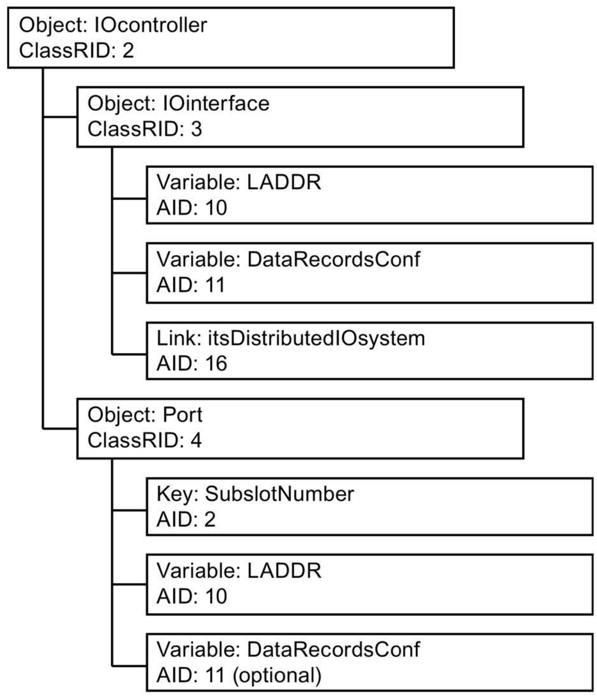

Figure 2-2 IOcontroller

Example of an IOcontroller

```
<Object Name="PN Driver">
     <ClassRID>2</ClassRID>
     <Object Name="PROFINET interface_1">
          <ClassRID>3</ClassRID>
          <Variable Name="LADDR">
              <AID>10</AID>
              ...............
          </Variable>
          <Variable Name="DataRecordsConf">
              <AID>11</AID>
              <Value>
                 ...............
              </Value>
          </Variable>
          <Link>
              <AID>16</AID>
              <TargetRID>12345</TargetRID>
          </Link>
      </Object>
      <Object Name="Port_1">
          <ClassRID>4</ClassRID>
          <Key AID="2">32769</Key>
          <Variable Name="LADDR">
               <AID>10</AID>
               ...............
          </Variable>
          <Variable Name="DataRecordsConf">
               <AID>11</AID>
               <Value>
                    ...............
               </Value>
           </Variable>
      </Object>
</Object>
```

2.5 distributedIOsystem

Overview of distributedIOsystem structure

The distributedIOsystem is structured as follows:

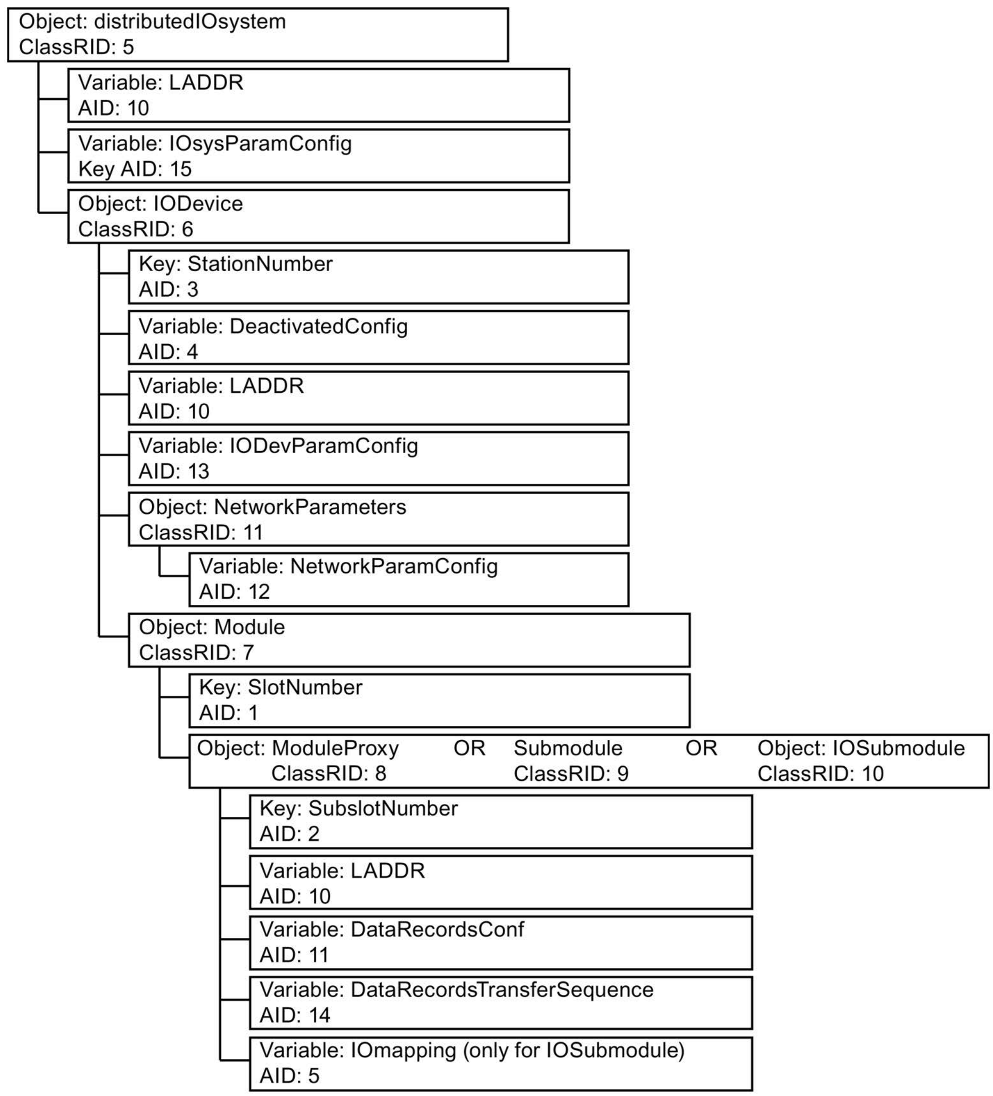

Figure 2-3 distributedIOsystem

Example of a distributedIOsystem with one IO device

```
<Object Name="PROFINET IO system">
     <RID>12345</RID>
     <ClassRID>5</ClassRID>
     <Variable Name="LADDR">
        <AID>10</AID>
        ...............
     </Variable>
     <Variable Name="IOsysParamConfig">
        <AID>15</AID>
        <Value>
          ...............
        </Value>
     </Variable>
     <Object Name="dev1">
        <GSDMLFile>GSDML-V2.31-Siemens-ET200SP-20150218.xml</GSDMLFile>
        <ClassRID>6</ClassRID>
        <Key AID="3">1</Key>
        <Variable Name="DeactivatedConfig">
           <AID>4</AID>
           <Value>
              ...............
           </Value>
        </Variable>
        <Variable Name="LADDR">
           <AID>10</AID>
           ...............
        </Variable>
        <Variable Name="IODevParamConfig">
           <AID>13</AID>
           <Value>
              ...............
           </Value>
        </Variable>
        <Object Name="Network Parameters">
           <ClassRID>11</ClassRID>
           <Variable Name="NetworkParamConfig">
              <AID>12</AID>
              <Value>
                 ...............
              </Value>
           </Variable>
         </Object>
         <Object Name="4DO x 24VDC / 2A ST_1">
            <ClassRID>7</ClassRID>
            <Key AID="1">0</Key>
            <Object Name="4DO x 24VDC / 2A ST_1">
               <ClassRID>10</ClassRID>
               <Key AID="2">1</Key>
               <Variable Name="LADDR">
                  <AID>10</AID>
                  ...............
```

```
            </Variable>
            <Variable Name="DataRecordsConf">
               <AID>11</AID>
               <Value>
                  ...............
               </Value>
            </Variable>
            <Variable Name="DataRecordsTransferSequence">
               <AID>14</AID>
                ...............
            </Variable>
            <Variable Name="IOmapping">
               <AID>5</AID>
               <Value>
               ...............
               </Value>
            </Variable>
          </Object>
        </Object>
      </Object>
   </Object>
```

2.6 PROFINET IO data records

PROFINET IO data records in XML elements

PROFINET data records are placed in XML elements of type **Variable**. The value of the XML attribute **Name** is either DataRecordsConf or NetworkParamConfig.

Each record is placed in the XML element **Field**. The XML attribute **Key** contains the record index and the attribute **Length** contains the record length.

The following is an example with one PROFINET IO data record. This record’s index is decimal 12304 (0x3010) and it is 12 bytes long. Record data must be in hexadecimal notation.

Example with one PROFINET IO data record

```
<Variable Name="DataRecordsConf">
    <AID>11</AID>
    <Value Datatype="SparseArray" Valuetype="BLOB">
       <Field Key="12304" Length="12">301400080100000000000000</Field>
    </Value>
</Variable>
```

2.7 Vendor-specific data records

Vendor-specific data records in XML elements

Some records of XML output are vendor-specific data records. These records contain data such as vendor identification data, feature-specific data, vendor-specific diagnostic, alarm and error data etc.

Vendor-specific data records are added to DataRecordsConf. The index of these data blocks is in the range of 0x0000-0x7FFF. Record data must be in hexadecimal notation.

In the following example, the index of the record is 0x0002 and it is 8 bytes long.

Example of a vendor-specific data record

```
<Variable Name="DataRecordsConf">
   <AID>11</AID>
   <Value Datatype="SparseArray" Valuetype="BLOB">
      <Field Key="2" Length="8">0801000000000000</Field>
   </Value>
</Variable>
```

2.8 GSDML file information

GSDML file information in XML elements

If any of these IO device objects are configured in TIA Portal using GSDML devices, then an additional "<GSDMLFile>" XML element is printed. If the device is not a GSDML device, this tag is not added. The inner text of this tag contains the GSDML file name for this device.

Example of GSDML file information

```
<Object Name="dev1">
   <GSDMLFile>GSDML-V2.31-Siemens-ET200SP-20150218.xml</GSDMLFile>
   <ClassRID>6</ClassRID><Object Name="dev1">
    ...........
```

PNIO classes, attributes and links 3

3.1 Classes

PNIO classes

The table below lists the PNIO classes that are available.

Table 3- 1 PNIO classes

Class	ClassRID number	Description
HWConfiguration	1	Container which aggregates all HW objects of the automation system
IOcontroller	2	IO controller
IOinterface	3	Submodule which acts as interface for distributed IO (applies to IOcontroller only)
Port	4	Submodule which acts as a port of an interface (applies to IOcontroller only)
distributedIOsystem	5	IO system which supports PROFINET IO
IODevice	6	IO device of a distributed IO system
Module	7	Module which acts only as a container for its submodules
ModuleProxy	8	Submodule which acts as module proxy, gets subslot number 0xFFF0
Submodule	9	Submodules are part of a module
IOSubmodule	10	Normal IO submodule to be aggregated at modules which carry IO. It is also used as module proxy if only one submodule is in the module.
NetworkParameters	11	Contains the network parameters as sets of data records. The NetworkParameters object is used only for the interfaces of the IOcon-troller.

3.2 Object attributes

PNIO object attributes

The table below lists the PNIO object attributes that are available.

Table 3- 2 PNIO object attributes

Attribute	AID number	Description	Type
SlotNumber	1	Slot number of the module, counts >=1	UINT32
SubslotNumber	2	Number of the subslot where the submodule is plugged	UINT32
StationNumber	3	Logical station number	UINT32
DeactivatedConfig	4	Configuration information on whether the HW object shall be deactivated at startup	BOOL
IOmapping	5	Specifies the IO addresses for input and output data	IOmapping
Ibase	6	Base byte address for inputs	UINT32
Ilength	7	Byte length of inputs	UINT16
Qbase	8	Base byte address for outputs	UINT32
Qlength	9	Byte length of outputs	UINT16
LADDR	10	HW identification (logical address)	UINT16
DataRecordsConf	11	Data records which are configured by the engineering system for the (sub)module	DataRecords
NetworkParamConfig	12	Network parameters which are configured by the engineering system for the IO device	BLOB_SP
IODevParamConfig	13	IO device parameters which are configured by the engineering system	BLOB_SP
DataRecordsTransferSequence	14	If present, this attribute defines the sequence according to which the configured data records must be transferred.	BLOB
IOsysParamConfig	15	Configures communication settings	BLOB_SP

3.3 Links

PNIO links

The table below lists the PNIO links that are available.

Table 3- 3 PNIO links

Link	Link ID number	Description	Target class
itsDistributedIOsystem	16	Controls the IO system as IO controller	distributedIOsystem

4 Used data types

Used data types

In the table below, the data types used for the description of the PNIO controller are listed.

Table 4- 1 Used data types

Type name	Description	Basic type
DataRecords	Map of all possible data records of a (sub)module	BLOB_SP
IOmapping	Contains complete IO address information of an IOSubmodule: • input base and length as byte addresses and length • output base and length as byte addresses and length	STRUCTURE OF Ibase: UINT32 // Base byte address for inputs Ilength: UINT16 // Byte length of inputs Qbase: UINT32 // Base byte address for out-puts Qlength: UINT16 // Byte length of outputs
BLOB	BLOB is an array of UINT8.	--
BLOB_SP	BLOB_SP is a map of BLOBs that are indexed by a qualifier.	--
BOOL	typedef bool BOOL; // 1 bit boolean #define FALSE false #define TRUE true	--
UINT32	Unsigned integer, 32 bits	--
UINT16	Unsigned integer, 16 bits	--

PNIO data blocks

5

5.1 PNIO data blocks header

PNIO data blocks header

BLOB data types are used for data blocks. This data is structured and all blocks described in this document have the following standard header:

Table 5- 1 Header for PNIO data blocks

Block Type	uint16_t	Unique block type
BlockLength	uint16_t	Length of block excluding BlockType and BlockLength (thus including BlockVersionHigh, BlockVersionLow, Reserved and the block data)
BlockVersionHigh	uint8_t	Block version major
BlockVersionLow	uint8_t	Block version minor
Reserved	uint16_t	Padding

Note

Vendor-specific data blocks and PROFINET IO data blocks may use a different header layout. These data blocks are distributed by PN Driver to the IO devices.

Each BLOB thus has a header with

- a BlockType, describing uniquely the content of the block,
- a BlockLength, allows multiple blocks to be stacked into one BLOB, among other things,
- a BlockVersion to identify the version of the block (as major and minor number),
- a Reserved to arrange the first item of the block on a 4 byte boundary.

The header and the content of a BLOB are always stored in big endian format.

5.2 Data blocks for IOInterface

Blocks for IOInterface

The following blocks can be assigned to an IOInterface of an IOcontroller:

Table 5- 2 Data blocks for IOInterface

Block	Block type	Index	Defined in section
IPV4_SUITE	0x3000	0x00001000	IPV4_SUITE (Page 28)
IP_ADDRESS_VALIDATION_LOCAL	0x3006	0x00001001	IP_ADDRESS_VALIDATION_LOCAL (Page 29)
NAME_OF_STATION	0xA201	0x00001003	NAME_OF_STATION (Page 29)
NAMEOFSTATION_VALIDATION	0x3009	0x00001004	NAMEOFSTATION_VALIDATION (Page 30)
SEND_CLOCK	0xF000	0x00010000	SEND_CLOCK (IOInterface) (Page 32)
PD_MASTER_TAILOR_DATA	0x7081	0x00017081	PD_MASTER_TAILOR_DATA (Page 34)
PN_IDENTIFICATION	0xF001	0x00023100	PN_IDENTIFICATION (Page 36)

5.3 Data blocks for network parameters

Blocks for Network Parameters (IODevice only)

The following blocks can be assigned to a NetworkParameters object, but they are only relevant for IO devices:

Table 5- 3 Data blocks for Network Parameters

Block	Block type	Index	Defined in section
IPV4_SUITE	0x3000	0x00001000	IPV4_SUITE (Page 28)
NAME_OF_STATION	0xA201	0x00001003	NAME_OF_STATION (Page 29)
STATION_NAME_ALIAS	0x3010	0x00001006	STATION_NAME_ALIAS (network parameter) (Page 31)
IP_ADDRESS_VALIDATION_REMOTE	0x3011	0x00001007	IP_ADDRESS_VALIDATION_REMOTE (network parameter) (Page 33)

5.4 Data blocks for IODevice

Blocks for IODevice

The following blocks can be assigned to an IODevice:

Table 5- 4 Data blocks for IODevice

Block	Block type	Index	Defined in section
AR_COMMUNICATION_DATA	0x3100	0x00003100	AR_COMMUNICATION_DATA (Page 37)
EXPECTED_SUBMODULE_DATA	0x3101	0x00003101	EXPECTED_SUBMODULE_DATA (Page 39)
IOCR_DATA	0x3102	0x00003102	IOCR_DATA (Page 41)
LOCAL_SCF_ADAPTION	0x3104	0x00003104	LOCAL_SCF_ADAPTION (Page 44)
ALARMCR_DATA	0x3107	0x00003107	ALARMCR_DATA (Page 45)
PNIOD_PROPERTIES	0x3060	0x00003060	PNIOD_PROPERTIES (Page 46)
AR_RECORD_DATA	0x3105	0x00003105	AR_RECORD_DATA (Page 48)

The following blocks can be assigned to an interface submodule of an IODevice:

Table 5- 5 Data blocks for interface submodule of IODevice

Block	Block type	Index	Defined in section
PD_MASTER_TAILOR_DATA	0x7081	0x00017081	PD_MASTER_TAILOR_DATA (Page 34)

5.5 Data blocks for distributedIOSystem

Blocks for distributedIOSystem

The following blocks can be assigned to a distributedIOsystem:

Table 5- 6 Data blocks for distributedIOSystem

Block	Block type	Index	Defined in section
CONTROLLER_PROPERTIES	0x3040	0x00003040	CONTROLLER_PROPERTIES (Page 49)

Description of PNIO data blocks 6

6.1 Data blocks for network parameters and IOInterface

6.1.1 IPV4_SUITE

Data block IPV4_SUITE (network parameters and IOInterface)

Data block IPV4_SUITE consists of the following attributes:

Table 6- 1 Data block IPV4_SUITE

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	IPV4_SUITE	0x3000	uint16_t
2	BlockLength	Length of block without BlockType, Block-Length	0x0010	uint16_t
4	BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6	Reserved	Padding	0x0000	uint16_t
8	IPAddress	IPv4 address	Byte array	uint8_t[4]
12	SubnetMask	IPv4 subnet mask	Byte array	uint8_t[4]
16	DefaultGateway	IPv4 default gateway (default router)	Byte array	uint8_t[4]

6.1.2 IP_ADDRESS_VALIDATION_LOCAL

Data block IP_ADDRESS_VALIDATION_LOCAL (network parameters and IOInterface)

Data block IP_ADDRESS_VALIDATION_LOCAL consists of the following attributes:

Table 6- 2 Data block IP_ADDRESS_VALIDATION_LOCAL

<table>
<tr><th>Address offset BYTE</th><th>Designation</th><th colspan="2">Content</th><th>Value range</th><th>Data type</th></tr>
<tr><td>0</td><td>BlockType</td><td colspan="2">IP_ADDRESS_VALIDATION_LOCAL</td><td>0x3006</td><td>uint16_t</td></tr>
<tr><td>2</td><td>BlockLength</td><td colspan="2">Length of block without BlockType, BlockLength</td><td>0x0008</td><td>uint16_t</td></tr>
<tr><td>4</td><td>BlockVersionHigh</td><td colspan="2">Major BlockVersion</td><td>0x01</td><td>uint8_t</td></tr>
<tr><td>5</td><td>BlockVersionLow</td><td colspan="2">Minor BlockVersion</td><td>0x01</td><td>uint8_t</td></tr>
<tr><td>6</td><td>Reserved</td><td colspan="2">Padding</td><td>0x0000</td><td>uint16_t</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">IPAddressValida-tion</td><td>0x0000</td><td>Use IPSuite
(IPSuite value → data record 0x1000)</td><td rowspan="2">0x0000 or 0xFFFF
others reserved</td><td rowspan="2">uint16_t</td></tr>
<tr><td>0xFFFF</td><td>IPSuite configured on site</td></tr>
<tr><td>10</td><td>Reserved</td><td colspan="2">Padding</td><td>0x0000</td><td>uint16_t</td></tr>
</table>

6.1.3 NAME_OF_STATION

Data block NAME_OF_STATION (network parameters and IOInterface)

Data block NAME_OF_STATION consists of the following attributes:

Table 6- 3 Data block NAME_OF_STATION

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	NAME_OF_STATION	0xA201	uint16_t
2	BlockLength	Length of block without BlockType, Block-Length		uint16_t
4	BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6	Reserved	Padding	0x0000	uint16_t
8	NameOfStationLength	Length of NameOfSta-tion	0x0001..0x00F0	uint16_t
10	Reserved	Padding	0x0000	uint16_t
12	NameOfStation	NameOfStation		uint8_t[NameOf Sta-tionLength]
12 + NameOf Station-Length	Reserved	Padding to 32 bit alignment	[0x00]	uint8_t[]

6.1.4 NAMEOFSTATION_VALIDATION

Data block NAMEOFSTATION_VALIDATION (network parameters and IOInterface)

Data block NAMEOFSTATION_VALIDATION consists of the following attributes:

Table 6- 4 Data block NAMEOFSTATION_VALIDATION

Address offset BYTE	Designation	Content		Value range	Data type
0	BlockType	NAMEOFSTATION_VALIDATION		0x3009	uint16_t
2	BlockLength	Length of block without BlockType, BlockLength		0x0008	uint16_t
4	BlockVersionHigh	Major BlockVersion		0x01	uint8_t
5	BlockVersionLow	Minor BlockVersion		0x01	uint8_t
6	Reserved	Padding		0x0000	uint16_t
8	IPAddressValida-tion	0x0000	Use NameOfStation (NoS value → data record 0x1003)	0x0000 or 0xFFFF others reserved	uint16_t
		0xFFFF	NameOfStation configured on site		
10	Reserved	Padding		0x0000	uint16_t

6.1.5 STATION_NAME_ALIAS (network parameter)

Data block STATION_NAME_ALIAS (network parameter)

Data block STATION_NAME_ALIAS consists of the following attributes:

Table 6- 5 Data block STATION_NAME_ALIAS

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	STATION_NAME_ALIAS	0x3010	uint16_t
2	BlockLength	Length of block without BlockType, Block-Length	0x0008	uint16_t
4	BlockVer-sionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVersion-Low	Minor BlockVersion	0x00	uint8_t
6	Reserved	Padding	0x0000	uint16_t
8	CountAlias	Number of alias entries		uint16_t
1st Alias Entry				
10	AliasBlock-Length	Length of the alias block, including block length		uint16_t
12	Alias-NameLength	Length of station name alias		uint8_t
13	AliasName	Station name alias of IODevice. Example: “port-001.my-iod”		uint8_t array [Ali-asNameLength]
13 + Alias-NameLength	Reserved	Padding to 8 bytes alignment	0x00	uint8_t array[n]
2nd Alias Entry				
...	...			

6.1.6 SEND_CLOCK (IOInterface)

Data block SEND_CLOCK (IOInterface)

Data block SEND_CLOCK consists of the following attributes:

Table 6- 6 Data block SEND_CLOCK

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	SEND_CLOCK	0xF000	uint16_t
2	BlockLength	Length of block without BlockType, Block-Length	0x0008	uint16_t
4	BlockVer-sionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVer-sionLow	Minor BlockVersion	0x00	uint8_t
6	SendClock-Factor	Send Clock Factor in multiples of 31.25 μs	1..128 [256, 512, 1024]	uint16_t
12	SendClock-Properties	SendClock is fixed, ReductionRatio is not adaptable. The SendClock is determined by the PDEV. An AR is accepted only if the SendClock of its IOCRs is equal to the current SendClock.	0x0003	uint16_t
16	Reserved		0x0000	uint16_t

Note

In contrast to most other blocks described in this manual, the SEND_CLOCK data record does **not** have the reserved two bytes after the BlockVersionLow.

6.1.7 IP_ADDRESS_VALIDATION_REMOTE (network parameter)

Data block IP_ADDRESS_VALIDATION_REMOTE (network parameter)

Data block IP_ADDRESS_VALIDATION_REMOTE consists of the following attributes:

Table 6- 7 Data block IP_ADDRESS_VALIDATION_REMOTE

<table>
<tr><th>Address offset BYTE</th><th>Designation</th><th colspan="2">Content</th><th>Value range</th><th>Data type</th></tr>
<tr><td>0</td><td>BlockType</td><td colspan="2">IP_ADDRESS_VALIDATION_REMOTE</td><td>0x3011</td><td>uint16_t</td></tr>
<tr><td>2</td><td>BlockLength</td><td colspan="2">Length of block without BlockType, BlockLength</td><td>0x0008</td><td>uint16_t</td></tr>
<tr><td>4</td><td>BlockVersionHigh</td><td colspan="2">Major BlockVersion</td><td>0x01</td><td>uint8_t</td></tr>
<tr><td>5</td><td>BlockVersionLow</td><td colspan="2">Minor BlockVersion</td><td>0x00</td><td>uint8_t</td></tr>
<tr><td>6</td><td>Reserved</td><td colspan="2">Padding</td><td>0x0000</td><td>uint16_t</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">Validation</td><td>0x0000</td><td>Apply IPSuite
(IPSuite value → data record 0x1000)</td><td rowspan="2">0x0000 or 0xFFFF
others reserved</td><td rowspan="2">uint16_t</td></tr>
<tr><td>0xFFFF</td><td>Use discovered IPSuite</td></tr>
<tr><td rowspan="3">10</td><td rowspan="3">Properties</td><td>Bit 0..3</td><td>Reserved</td><td rowspan="3"></td><td rowspan="3">Bitfield[16]</td></tr>
<tr><td>Bit 4</td><td>0: set IP suite temporary
1: set IP suite permanent</td></tr>
<tr><td>Bit 5..15</td><td>Reserved</td></tr>
</table>

6.1.8 PD_MASTER_TAILOR_DATA

Data block PD_MASTER_TAILOR_DATA (tailoring parameters of IOInterface)

Data block PD_MASTER_TAILOR_DATA consists of the following attributes:

Table 6- 8 Data block PD_MASTER_TAILOR_DATA

Address offset BYTE	Designation		Content	Value range	Data type
0	BlockType		PD_MASTER_TAILOR_DATA	0x7081	uint16_t
2	BlockLength		Length of the entire structure block without BlockType, BlockLength		uint16_t
4	BlockVersionHigh		Major BlockVersion	0x01	uint8_t
5	BlockVersionLow		Minor BlockVersion	0x00	uint8_t
6	DeviceTailorProperties				Bitfield[16]
	0	DeviceMode	0: Device is mandatory 1: Device is optional		Bitfield[1]
	1	NameOfSta-tion	0: No tailoring of NameOfStation need-ed 1: Tailor NameOfStation		Bitfield[1]
	2	IPv4Suite	0: No tailoring of IPv4Suite needed 1: Tailor IPv4Suite		Bitfield[1]
	3	Reserved	Reserved	0	Bitfield[1]
	4	PDInter-faceMrpData Adjust	0: No tailoring of PDInter-faceMrpDataAdjust needed 1: Tailor PDInterfaceMrpDataAdjust		Bitfield[1]
	5	PDInter-faceMrpData Check	0: No tailoring of PDInter-faceMrpDataCheck needed 1: Tailor PDInterfaceMrpDataCheck		Bitfield[1]
	6.. 15	Reserved	Reserved	0	Bitfield[10]
8	DeviceGroup		Number of the device group the device belongs to		uint16_t
10	NumberOfPort-TailorInfos		Number of items of port tailor infor-mation which follow		uint16_t
1st Port Tailor Info					
12	LocalSlotNumber		Slot number of local port which needs to be tailored		uint16_t
14	LocalSubslotNum-ber		Subslot number of local port which needs to be tailored		uint16_t

<table>
<tr><th>Address offset BYTE</th><th colspan="2">Designation</th><th>Content</th><th>Value range</th><th>Data type</th></tr>
<tr><td>16</td><td colspan="3">PortTailorProperties</td><td></td><td>Bitfield[16]</td></tr>
<tr><td rowspan="5"></td><td>0-1</td><td>PeerMode</td><td>0x00: Fixed peer to mandatory neighbor
0x01: Fixed peer to optional neighbor
0x02: Programmable peer
0x03: Any peer</td><td></td><td>Bitfield[2]</td></tr>
<tr><td>2</td><td>NameOfPortCoding</td><td>0: NameOfPort coding is “port-xyz”
1: NameOfPort coding is “port-xyz-rstuv”</td><td></td><td>Bitfield[1]</td></tr>
<tr><td>3</td><td>Reserved</td><td>Reserved</td><td>0</td><td>Bitfield[1]</td></tr>
<tr><td>4</td><td>TailorPDPortMrpDataAdjust</td><td>0: No tailoring of PDPortMrpDataAdjust needed
1: Tailor PDPortMrpDataAdjust</td><td></td><td>Bitfield[1]</td></tr>
<tr><td>5..15</td><td>Reserved</td><td>Reserved</td><td>0</td><td>Bitfield[11]</td></tr>
<tr><td>18</td><td colspan="2">PeerStationNumber</td><td>Station number of the neighbor.
Must be set to 0xFFFF, if PeerMode is set to 0x02 or 0x03</td><td></td><td>uint16_t</td></tr>
<tr><td>20</td><td colspan="2">PeerSlotNumber</td><td>Slot number of the neighbor.
Must be set to 0xFFFF, if PeerMode is set to 0x02 or 0x03</td><td></td><td>uint16_t</td></tr>
<tr><td>22</td><td colspan="2">PeerSubslotNumber</td><td>Subslot number of the neighbor.
Must be set to 0xFFFF, if PeerMode is set to 0x02 or 0x03</td><td></td><td>uint16_t</td></tr>
<tr><td colspan="6">2nd Port Tailor Info</td></tr>
<tr><td>...</td><td colspan="2">...</td><td></td><td></td><td></td></tr>
</table>

Note

In contrast to most other blocks described in this manual, the PD_MASTER_TAILOR_DATA data record does **not** have the reserved two bytes after the BlockVersionLow.

6.1.9 PN_IDENTIFICATION

Data block PN_IDENTIFICATION

Data block PN_IDENTIFICATION consists of the following attributes:

Table 6- 9 Data block PN_IDENTIFICATION

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	PN_IDENTIFICATION	0xF001	uint16_t
2	BlockLength	Length of block without BlockType, BlockLength	0x0010	uint16_t
4	BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6	Reserved	Padding	0x0000	uint16_t
8	VendorID	Vendor ID of device		uint16_t
10	DeviceID	Device ID of device		uint16_t
12	InstanceID	Instance ID of device		uint16_t
14	Reserved	Reserved	0x00	uint8_t[6]

6.2 Data blocks for IODevice

6.2.1 AR_COMMUNICATION_DATA

Data block AR_COMMUNICATION_DATA V1.0 (IODevice)

Data block AR_COMMUNICATION_DATA V1.0 consists of the following attributes:

Table 6- 10 Data block AR_COMMUNICATION_DATA V1.0

Address offset BYTE	Designation		Content	Value range	Data type
0	BlockType		AR_COMMUNICATION_DATA	0x3100	uint16_t
2	BlockLength		Length of the entire structure block without BlockType, Block-Length	0x001C	uint16_t
4	BlockVersionHigh		Major BlockVersion	0x01	uint8_t
5	BlockVersionLow		Minor BlockVersion	0x00	uint8_t
6	Reserved			0x0000	uint16_t
8	ARBlockVersion		High_Byte (Major: 0x01) Low_Byte (Minor: 0x00) (The first version is 1.0)	01 00	uint16_t
10	ARType		Possible values: 0x0001 = IOCARSingle	0x0001	uint16_t
12	AR_UUID		UUID Has to be created by the config-uration		
	0	Data1			uint32_t
	4	Data2			uint16_t
	6	Data3			uint16_t
	8	Data4			uint8_t
	9	Data5			uint8_t
	10	Data6			uint8_t[6]
28	ARProperties		Properties of AR	Default: 0x00000011	Bitfield[32]
	0..2	Reserved	Reserved	1	Bitfield[3]
	3	Reserved	Reserved	0	Bitfield[1]
	4	Reserved	Reserved	1	Bitfield[1]
	5..29	Reserved	Reserved	0	Bitfield[25]
	30	StartupMode	Startup mode for RT (depending on the requested IOCRs). 0: Legacy mode 1: Advanced mode	Default: 0	Bitfield[1]
	31	Reserved	Reserved	0	Bitfield[1]

Data block AR_COMMUNICATION_DATA V1.1 (IODevice)

Data block AR_COMMUNICATION_DATA V1.1 consists of the following attributes:

Table 6- 11 Data block AR_COMMUNICATION_DATA V1.1

Address offset BYTE	Designation		Content	Value range	Data type
0	BlockType		AR_COMMUNICATION_DATA	0x3100	uint16_t
2	BlockLength		Length of the entire structure block without BlockType, Block-Length	0x003C	uint16_t
4	BlockVersionHigh		Major BlockVersion	0x01	uint8_t
5	BlockVersionLow		Minor BlockVersion	0x01	uint8_t
6	Reserved			0x0000	uint16_t
8	ARBlockVersion		High_Byte (Major: 0x01) Low_Byte (Minor: 0x01)	01 01	uint16_t
10	ARType		Possible values: 0x0001 = IOCARSingle	0x0001	uint16_t
12	AR_UUID		UUID Has to be created by the config-uration		
	0	Data1			uint32_t
	4	Data2			uint16_t
	6	Data3			uint16_t
	8	Data4			uint8_t
	9	Data5			uint8_t
	10	Data6			uint8_t[6]
28	ARProperties		Properties of AR	Default: 0x00000011	Bitfield[32]
	0..2	Reserved	Reserved	1	Bitfield[3]
	3	Reserved	Reserved	0	Bitfield[1]
	4	Reserved	Reserved	1	Bitfield[1]
	5..29	Reserved	Reserved	0	Bitfield[25]
	30	StartupMode	Startup mode for RT (depending on the requested IOCRs). 0: Legacy mode 1: Advanced mode	Default: 0	Bitfield[1]
	31	Reserved	Reserved	0	Bitfield[1]
32	CMIActivityTimeout		Ramp up monitoring time in 100 ms, IO device monitors the IO controller. Must not be bigger than the RPC remote application	For legacy startup mode: 600 For advanced startup mode: 200	uint16_t
34..63	Reserved		Reserved	0	uint8_t[30]

6.2.2 EXPECTED_SUBMODULE_DATA

Data block EXPECTED_SUBMODULE_DATA (IODevice)

Data block EXPECTED_SUBMODULE_DATA consists of the following attributes:

Table 6- 12 Data block EXPECTED_SUBMODULE_DATA

Address offset BYTE			Designation	Content	Value range	Data type
0			BlockType	EXPECTED_SUBMODULE_ DATA	0x3101	uint16_t
2			BlockLength	Length of the entire structure block without BlockType, BlockLength	2..65535	uint16_t
4			BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5			BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6			Reserved		0x0000	uint16_t
8			Number of APIs	Number of following API blocks	0..65536	uint16_t
API 1 Entry						
	0		API		$0..2^{32}$	uint32_t
	4		Reserved	Reserved	0	uint16_t
	6		Number of slot blocks		0..65536	uint16_t
		Slot 1 entry				
		0	Slot block length			uint16_t
		2	SubmoduleDataBlockVersion	High_Byte (Major: 0x01) Low_Byte (Minor: 0x00) (The first, official version should be 1.0.)	01 00	uint16_t
		4	SlotNumber	0..0xFFFF		uint16_t
		6	Reserved		0	uint16_t
		8	ModuleIdentNumber	Configured ID		uint32_t
		12	Reserved		0	uint16_t
		14	MaxSubmoduleNumber	Highest configured PNIO submodule number of mod-ule regarding API [1]	1..65535	uint16_t
		16	Reserved		0	uint16_t
		18	NumberOfSubmodule De-scriptions	Number of submodule de-scription blocks of this AR		uint16_t

Address offset BYTE					Designation	Content		Value range	Data type
			Submodule description 1 entry						
			0		SubslotNumber	PNIO submodule number; in ascending order from 1..0xFFFE			uint16_t
			2		Reserved			0	uint16_t
			4		SubmoduleIdentNumber	Configured ID			uint32_t
			8		Submodule Properties:				Bitfield[16]
				0..1	Submodule-Type	0	Submodule contains no Input_Data and no Output_Data [submodules without working data] (only one input submodule data description with length 0 follows)		Bitfield[2]
						1	Submodule contains only Input_Data (one input submodule data description block follows)		
						2	Submodule contains only Output_Data (one output submodule data description block follows)		
						3	Submodule contains Output_Data and Input_Data (two submodule data description blocks, one for the inputs and one for the outputs, follow)		
				2..15	Reserved			0	Bitfield[14]
			10		Reserved			0	uint16_t
			X		Submodule data description (1 or 2):				
				0	TypeOfDataDescription	0	Output data description follows[2]		Bitfield[1]
						1	Input data description follows		
				1..15	Reserved			0	Bitfield[15]
			X+2		Length Data	Length of the working data		0..1439	uint16_t
			X+4		Length IOPS	Length of the IOPS		1	uint16_t
			X+6		Length IOCS	Length of the IOCS		1	uint16_t
			X+8		Reserved			0	uint32_t
			Submodule description n entry						
				...					
		Slot n entry							
			...						

Address offset BYTE		Designation	Content	Value range	Data type
API n entry					
		...			
			Padding to 16 bytes alignment		

1 The highest configured submodule number regarding the API has to be entered here. Example: The module contains the following submodules regarding the API: 1, 10, 100; hence MaxSubmoduleNumber = 100.

2 Deviation from IEC 61158 Standard. In IEC 61158 value of Output is 0x02.

6.2.3 IOCR_DATA

Data block IOCR_DATA (IODevice)

Data block IOCR_DATA consists of the following attributes:

Table 6- 13 Data block IOCR_DATA

Address offset BYTE		Designation	Content	Value range	Data type
0		BlockType	IOCR_DATA	0x3102	uint16_t
2		BlockLength	Length of the entire structure block without BlockType, Block-Length	2..65535	uint16_t
4		BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5		BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6		Reserved		0x0000	uint16_t
8		Number of CRs			uint16_t
	IO CR entry 1				
	0	Block length	Length of the block including block length [2..65535]		uint16_t
	2	IOCRBlockVersion	High_Byte (Major: 0x01) Low_Byte (Minor: 0x00) (The first, official version should be 1.0.)	01 00	uint16_t
	4	IOCRType	Possible values: 0x0001 = Input CR 0x0002 = Output CR Rest = reserved		uint16_t
	6	IOCRReference	As the IO data objects are distributed by the configuration to the CRs, a reference between CR and module/submodule description is needed.		uint16_t
	8	Reserved		0	uint32_t

Address offset BYTE		Designation				Content	Value range	Data type
	12	IOCRProperties:						Bitfield[32]
		0..3	RTClass			1= Use FrameID range 7 for the IOCRs used for RT_CLASS_1 (legacy) Rest reserved	1..2	Bitfield[4]
		4..31	Reserved				0	Bitfield[28]
	16	DataLength				Genuine working data length of the CR	0..1440	uint16_t
	18	FrameID				RTClass ==_1 (Unicast) legacy: FrameID must be a unique number from FrameID range 7.		uint16_t
	20	SendClockFactor				TimeBase: 31.25 µs SendClock:= SendClockFactor * TimeBase	1..128 default 32	uint16_t
	22	ReductionRatio				Mandatory value range: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 Reserved value range: 0, >512 Optional value range: the rest	1..512	uint16_t
	24	Phase				Selected time section Permitted range: 0 = Reserved 1.."ReductionRatio"		uint16_t
	26	Reserved					0	uint16_t
	28	Reserved					0	uint32_t
	32	Reserved					0	uint16_t
	34	DataHoldFactor				DataHoldTime := DataHoldFactor * SendClockFactor * Reduction-Ratio * 31.25 µs Value range from 3..0x1E00 for RT Class 1; the DataHoldTime shall be equal to or less than 1.92 s.	default 3	uint16_t
	36	Reserved					0	uint16_t[11]
	58	Number of APIs					0..65536	uint16_t
		API entry 1						
		0	API				$0..2^{32}$	uint32_t
		4	NumberOfRelatedIO-DataObjects			Depending on the CR, this field is either for input data or output data		uint16_t
			0	SlotNumber			1..32768	uint16_t
			2	Subslotnumber			1..32768	uint16_t
			4	FrameOffset:			0..1439	Bitfield[16]
				0..10	Offset			Bitfield[11]
				11..15	Reserved		0	Bitfield[5]

Address offset BYTE		Designation				Content	Value range	Data type
			6	Reserved			0	uint16_t
			Reserved				0	uint16_t
			NumberOfRelatedIOCS			Depending on the CR, this field is either for input data or output data		uint16_t
			0	SlotNumber			1..32768	uint16_t
			2	Subslotnumber			1..32768	uint16_t
			4	FrameOffset:			0..1439	Bitfield[16]
				0..10	Offset			Bitfield[11]
				11..15	Reserved		0	Bitfield[5]
			6	Reserved			0	uint16_t
			Reserved				0	uint16_t
		API entry n						
		N	API					
			..					
	IO CR entry n							
		...						
						Padding to 16 bytes alignment		

6.2.4 LOCAL_SCF_ADAPTION

Data block LOCAL_SCF_ADAPTION (IODevice)

Data block LOCAL_SCF_ADAPTION consists of the following attributes:

Table 6- 14 Data block LOCAL_SCF_ADAPTION

<table>
<tr><th colspan="2">Address offset BYTE</th><th>Designation</th><th>Content</th><th>Value range</th><th>Data type</th></tr>
<tr><td>0</td><td></td><td>BlockType</td><td>LOCAL_SCF_ADAPTION</td><td>0x3104</td><td>uint16_t</td></tr>
<tr><td>2</td><td></td><td>BlockLength</td><td>Length of the entire structure block without BlockType, BlockLength</td><td>2..65535</td><td>uint16_t</td></tr>
<tr><td>4</td><td></td><td>BlockVersionHigh</td><td>Major BlockVersion</td><td>0x01</td><td>uint8_t</td></tr>
<tr><td>5</td><td></td><td>BlockVersionLow</td><td>Minor BlockVersion</td><td>0x00</td><td>uint8_t</td></tr>
<tr><td>6</td><td></td><td>Reserved</td><td></td><td>0x0000</td><td>uint16_t</td></tr>
<tr><td>8</td><td></td><td>Number of local SCF adaption entries</td><td></td><td>0..x</td><td>uint16_t</td></tr>
<tr><td rowspan="13"></td><td colspan="5">Local SCF adaption entry 1</td></tr>
<tr><td>0</td><td>Block length</td><td>Length of the block including block length</td><td>20</td><td>uint16_t</td></tr>
<tr><td>2</td><td>IOCRReference</td><td>Reference to corresponding entry in the IOCR data table</td><td></td><td>uint16_t</td></tr>
<tr><td>4</td><td>LocalSendClockFactor</td><td>LocalSendClockFactor covers the range from 1..1024. Increment is 1.
LocalSendClock := LocalSendClock-Factor * 31.25 μs</td><td>1..1024</td><td>uint16_t</td></tr>
<tr><td>6</td><td>LocalReductionRatio</td><td>Mandatory value range: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512
Reserved value range: 0, >512
Optional value range: the rest</td><td>1..512</td><td>uint16_t</td></tr>
<tr><td>8</td><td>LocalPhase</td><td>Selected time section
Permitted range:
0 = reserved
1.."LocalReductionRatio"</td><td></td><td>uint16_t</td></tr>
<tr><td>10</td><td>Reserved</td><td></td><td>0</td><td>uint16_t</td></tr>
<tr><td>12</td><td>Reserved</td><td></td><td>0</td><td>uint32_t</td></tr>
<tr><td>16</td><td>Reserved</td><td></td><td>0</td><td>uint16_t</td></tr>
<tr><td>18</td><td>LocalDataHoldFactor</td><td>LocalDataHoldTime := Local-DataHoldFactor * SendClockFactor * ReductionRatio * 31.25 μs
Value range from 3..0x1E00 for RT Class 1; the LocalDataHoldTime shall be equal to or less than 1.92 s.
Special case “0”: Use the value DataHoldFactor from the IOCR Data (see section IOCR_DATA (Page 41))</td><td></td><td>uint16_t</td></tr>
<tr><td colspan="5">Local SCF adaption entry x</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2"></td><td>Reserved</td><td>Padding to 32 bit alignment</td><td>0x0000</td><td>uint16_t</td></tr>
</table>

6.2.5 ALARMCR_DATA

Data block ALARMCR_DATA (IODevice)

Data block ALARMCR_DATA consists of the following attributes:

Table 6- 15 Data block ALARMCR_DATA

Address offset BYTE	Designation	Content	Value range	Data type
0	BlockType	ALARMCR_DATA	0x3107	uint16_t
2	BlockLength	Length of the entire structure block without BlockType, Block-Length	0x0014	uint16_t
4	BlockVersionHigh	Major BlockVersion	0x01	uint8_t
5	BlockVersionLow	Minor BlockVersion	0x00	uint8_t
6	Reserved		0x0000	uint16_t
8	AlarmCRVersion	High_Byte (Major: 0x01) Low_Byte (Minor: 0x00)	01 00	uint16_t
10	AlarmCRType	0x0001 = AlarmCR	0x0001	uint16_t
12	Ethertype		0x8892	uint16_t
14	Reserved		0	uint16_t
16	Reserved		0	uint32_t
20	RTATimeoutFactor	TimeBase 100 ms RTATimeout := RTATimeoutFactor * TimeBase	1..100 Default: 1	uint16_t
22	RTARetries	Repeating counter (1..15)	1..15 Default: 3	uint16_t
24	AlarmCRTagHead-erHigh		0xC000	Bitfield[16]
26	AlarmCRTagHeader-Low		0xA000	Bitfield[16]

6.2.6 PNIOD_PROPERTIES

Data block PNIOD_PROPERTIES (IODevice)

Data block PNIOD_PROPERTIES consists of the following attributes:

Table 6- 16 Data block PNIOD_PROPERTIES

Address offset BYTE	Designation			Content	Value range	Data type
0	BlockType			PNIOD_PROPERTIES	0x3060	uint16_t
2	BlockLength			Length of the entire structure block without BlockType, BlockLength	0x001C	uint16_t
4	BlockVersionHigh			Major BlockVersion	0x01	uint8_t
5	BlockVersionLow			Minor BlockVersion	0x00	uint8_t
6	Reserved				0x0000	uint16_t
8	Configured share of the object UUID of the IO device:					
	0	Vendor ID		Vendor ID		uint16_t
	2	Device ID		Device ID		uint16_t
	4	Instance ID		Instance ID		Bitfield[16]
		0	InstanceNumber	Represents the instance or node number		Bitfield[12]
		12	InterfaceNumber	Identifies the interface		Bitfield[4]
14	MaxRecordSize			Maximum size of a data record	1..65535	uint16_t
16	DeviceProperties					Bitfield[32]
	0	MultipleWriteSupported		0: Multiple Write not supported 1: Multiple Write supported	0..1	Bitfield[1]
	1..31	Reserved		Reserved	0	Bitfield[31]

Address offset BYTE	Designation		Content	Value range	Data type
20	DeviceMode				Bitfield[32]
	0..7	HighPrioScanCycleIn10ms	Scan cycle for priority devic-es (TimeBase: 10 ms) 0: Use low priority scan cycle 50: Use 500 ms scan cycle (for accelerated startup procedure)	0, 50	Bitfield[8]
	8	Reserved	Reserved	0	Bitfield[1]
	9	FastStartupProcedure	0: Use simple procedure 1: Use fast startup proce-dure	0, 1	Bitfield[1]
	10	AcceleratedStartupProcedure	0: Use simple procedure 1: Use accelerated startup procedure	0, 1	Bitfield[1]
	11	Reserved	Reserved	0	Bitfield[1]
	12	CheckDeviceID	0: Do not check VendorID/ DeviceID/ InstanceID 1: Check VendorID/ De-viceID/ InstanceID	0, 1	Bitfield[1]
	13	AllowNameOfStationOverwrite	0: Overwriting device’s NameOfStation is not al-lowed 1: Overwriting device’s NameOfStation is allowed	0, 1	Bitfield[1]
	14..31	Reserved	Reserved	0	Bitfield[18]
24..31	Reserved			0	uint16_t[4]

6.2.7 AR_RECORD_DATA

Data block AR_RECORD_DATA (IODevice)

Data block AR_RECORD_DATA consists of the following attributes:

Table 6- 17 Data block AR_RECORD_DATA

<table>
<tr><th>Address offset BYTE</th><th colspan="2">Designation</th><th>Content</th><th>Value range</th><th>Data type</th></tr>
<tr><td>0</td><td colspan="2">BlockType</td><td>AR_RECORD_DATA</td><td>0x3105</td><td>uint16_t</td></tr>
<tr><td>2</td><td colspan="2">BlockLength</td><td>Length of the entire structure block without BlockType, BlockLength</td><td>2..65535</td><td>uint16_t</td></tr>
<tr><td>4</td><td colspan="2">BlockVersionHigh</td><td>Major BlockVersion</td><td>0x01</td><td>uint8_t</td></tr>
<tr><td>5</td><td colspan="2">BlockVersionLow</td><td>Minor BlockVersion</td><td>0x00</td><td>uint8_t</td></tr>
<tr><td>6</td><td colspan="2">Reserved</td><td>Reserved</td><td>0x0000</td><td>uint16_t</td></tr>
<tr><td>8</td><td colspan="2">NumberofRecords</td><td>Number of records</td><td>0x01</td><td>uint16_t</td></tr>
<tr><td colspan="6">1st data record</td></tr>
<tr><td rowspan="6"></td><td>0</td><td>Block length</td><td></td><td></td><td>uint16_t</td></tr>
<tr><td>2</td><td>Data record length</td><td>Length of the data record</td><td></td><td>uint16_t</td></tr>
<tr><td>4</td><td>Data record index</td><td>Index of record</td><td>0xE000..0xEFFF</td><td>uint16_t</td></tr>
<tr><td>6</td><td>Reserved</td><td>Reserved</td><td>0</td><td>uint16_t</td></tr>
<tr><td>8</td><td>Parameter data</td><td>Contains the content of the record data object</td><td></td><td>uint8_t[]</td></tr>
<tr><td>8+ x</td><td>Reserved</td><td>Padding to 8 bytes alignment</td><td></td><td>uint8_t[]</td></tr>
<tr><td colspan="6">2nd data record</td></tr>
<tr><td></td><td>...</td><td>...</td><td></td><td></td><td></td></tr>
</table>

6.3 Data blocks for distributedIOSystem

6.3.1 CONTROLLER_PROPERTIES

Data block CONTROLLER_PROPERTIES (distributedIOSystem)

Data block CONTROLLER_PROPERTIES consists of the following attributes:

Table 6- 18 Data block CONTROLLER_PROPERTIES

Address offset BYTE	Designation				Content	Value range	Data type
0	BlockType				CONTROLLER_PROPERTIES	0x3040	uint16_t
2	BlockLength				Length of the entire structure block without BlockType, Block-Length	0x0010	uint16_t
4	BlockVersionHigh				Major BlockVersion	0x01	uint8_t
5	BlockVersionLow				Minor BlockVersion	0x00	uint8_t
6	Reserved					0x0000	uint16_t
8	Configured share of the object UUID of the IO controller:						
	0	Vendor ID			Vendor ID		uint16_t
	2	Device ID			Device ID		uint16_t
	4	Instance ID			Instance ID		Bitfield[16]
		0	InstanceNumber		Represents the in-stance or node num-ber		Bitfield[12]
		12	InterfaceNumber		Identifies the interface		Bitfield[4]
14	CMIActivityTimeout				Ramp up monitoring time in 100 ms, IO device monitors the IO controller Must not be bigger than the RPC Remote Application Timeout	600	uint16_t
16	RPC Remote Application Timeout				Timeout for RPC calls in [1s] On expiration, the IO controller/IO device cuts off the RPC call.	300	uint16_t
18	Reserved				Padding to 32 bit alignment	0x0000	uint16_t

Appendix A

A.1 Abbreviations / Glossary of terms

AID	**A**ttribute **Id**entifier
API	**A**pplication **P**rocess **I**dentifier (profile) or **A**pplication **P**rogramming **I**nterface
AR	**A**pplication **R**elation
BLOB	**B**inary **L**arge **Ob**ject
IO	**I**nput/**O**utput
IOC	**IO** **C**ontroller
IOCR	**I**nput/**O**utput **C**ommunication **R**elationship
IOCS	**I**nput/**O**utput **C**onsumer **S**tatus
IOPS	**I**nput/**O**utput **P**rovider **S**tatus
LADDR	**L**ogical Base **Addr**ess
PDEV	**P**hysical **Dev**ice
PNIO	**P**ROFI**N**ET **IO**
PNIOC	**P**ROFI**N**ET **IO** **C**ontroller
PNIOD	**P**ROFI**N**ET **IO** **D**evice
RID	**R**untime **Id**entifier
RT	**R**eal-**t**ime is a generic term for acyclic and cyclic real-time communication
UUID	**U**niversal **U**nique **Id**entifier
XML	E**x**tensible **M**arkup **L**anguage