

IO-Link Gateway Management

Specification

Draft for PI-Review

**D1.0.0-01
August 2026**

Order No: 10.192

File name: IOL-Gateway-Management_10192_d100_Aug26

This specification has been prepared by the IO-Link Gateway management team and released by the IO-Link community as draft for review version.

Any comments, proposals, requests on this document are appreciated. Please use www.io-link-projects.com for your entries and provide name and email address.

Login: **IOL-Gateway**

Password: **Report**

NOTICE:

The information contained in this document is subject to change without notice. The material in this document details a PNO specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of a PNO specification in any company's products.

The attention of adopters is directed to the possibility that compliance with or adoption of PNO specifications may require use of an invention covered by patent rights. PNO shall not be responsible for identifying patents for which a license may be required by any PNO specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. PNO specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

WHILE THE INFORMATION IN THIS DOCUMENT IS BELIEVED TO BE ACCURATE, PNO MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event shall PNO be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with PNO specifications does not absolve manufacturers, from the requirements of safety and regulatory agencies (TÜV, BIA, UL, CSA, etc.).

USE OF TRADEMARKS:

☞ **IO-Link** ® is registered trade mark. The use is restricted for members of the IO-Link Community. More detailed terms for the use can be found in the IO-Link Community Rules on www.io-link.com.

PNO is the owner of several registered trademarks, such as PROFIBUS®, PROFINET®, omlox®, IO-Link®, MTP® and others. More detailed terms for the use can be found on the web page www.profibus.com. Please select buttons "Downloads / Presentations & logos". In some cases, PNO is the licensee of registered trademarks owned by third parties and which may be relevant in regard with products compliant to this document.

PNO shall always be the sole entity that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with a PNO specification. Products developed using a PNO specification may claim compliance or conformance with a PNO specification only if the hardware and/or software satisfactorily meets the certification requirements set by PNO. Products that do not meet these requirements may claim only that the product was based on a PNO specification and must not claim compliance or conformance with a PNO specification.

COPYRIGHT

Copyright © 2026 PROFIBUS Nutzerorganisation e.V.

Any unauthorized use of this publication may violate Copyright Law, Trademark Law and other legal regulations. This document contains information which is protected by Copyright. No part of this work covered by Copyright herein may be reproduced or used in any form or by any means -graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the publisher.

Publisher:

IO-Link Community

c/o PROFIBUS Nutzerorganisation e.V.

Ohiostrasse 8

76149 Karlsruhe

Germany

Phone: +49 721 / 98 61 97 0

Fax: +49 721 / 98 61 97 11

E-mail: info@io-link.com

Web site: www.io-link.com

Draft for Executive Review. Do not Claim Conformance!

LICENSE AGREEMENT

1. License

- 1.1 Subject of this license agreement is this document issued by the Licensor, in electronic form. If applicable, also software may be provided.
- 1.2 The user of this document (Licensee) acquires the license solely from PROFIBUS Nutzerorganisation e.V., having its principal place of business in Karlsruhe, Germany (hereinafter referred to as "Licensor").
- 1.3 This document is not an industrial standard acknowledged by any standardization body or otherwise and may be further enhanced.

2. Rights and Duties of Licensee

- 2.1 Licensor hereby grants to Licensee the right to use this document exclusively for developing and supporting products compliant with this document. Licensee may copy this document for this purpose and for data backup purposes.
- 2.2 Licensee shall not be entitled to modify, decompile, reverse engineer or extract any individual parts of this document, unless this is permitted by mandatory Copyright Law. Furthermore, Licensee shall not be entitled to remove any alphanumeric identifiers, trademarks or copyright notices from this document and, insofar as Licensee is entitled to make copies of this document, Licensee shall copy them without alteration.
- 2.3 Licensee is not entitled to copy and redistribute this document to any third party, except for "Have Made" purposes. All copies must be obtained on an individual basis, directly from the website www.de.profibus.com or www.profibus.com or upon request from the Licensor.
- 2.4. Licensee agrees to comply with all applicable export control, trade, and sanctions regulations as well as embargo regulations of the European Union, the United States of America, and other relevant jurisdictions ("Export Control Regulations"). This applies in particular to the applicable Export Control Regulations relating to Russia and Belarus and with natural or legal persons associated with Russia and Belarus. Should Licensor become aware of any violation of Export Control Regulations with respect to the use of this document, PNO reserves the right to terminate this Agreement without notice and claim damages.

3. Liability of Licensor

- 3.1 Licensor shall have no obligation to enhance the document and shall assume no liability in case the document or future versions thereof shall not be approved as an industrial standard.
- 3.2 Licensor's liability for defects as to quality or title of this document, especially in relation to the correctness or absence of defects or the absence of claims or third-party rights or in relation to completeness, usability and/or fitness for purpose are excluded, except for cases involving gross negligence, willful misconduct or fraudulent concealment of a defect.
- 3.3 Any further liability is excluded unless required by law, e.g. in cases of personal injury or death, willful misconduct, gross negligence, or in case of breach of fundamental contractual obligations. The damages in case of breach of fundamental contractual obligations is limited to the contract-typical, foreseeable damage if there is no willful misconduct or gross negligence.

4. Place of Jurisdiction and Applicable Law

- 4.1 The sole place of jurisdiction shall be the principal place of business of Licensor.
- 4.2 All relations arising out of the contract shall be governed by the substantive law of Germany, to the exclusion of the United Nations Convention on Contracts for the International Sale of Goods (CISG).

Conventions:

In this specification the following key words (in **bold** text) will be used:

shall:	indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interoperability and to claim conformity with this specification.
should:	indicates flexibility of choice with a strongly preferred implementation.
can:	indicates flexibility of choice with no implied preference (possibility and capability).
may:	indicates a permission.
highly recommended:	indicates that a feature shall be implemented except for well-founded cases. Vendor shall document the deviation within the user manual and within the manufacturer declaration.

CONTENTS

1	Introduction	7
1.1	Scope	7
1.2	List of affected patents.....	7
2	References	7
3	Terms, definitions, symbols, abbreviated terms and conventions	7
3.1	Terms and definitions.....	7
3.2	Symbols and abbreviated terms	8
4	Involved Technologies	8
4.1	IO-Link technology	8
4.2	Integration technologies	9
4.2.1	General	9
4.2.2	Classic fieldbus-based automation hierarchy	9
4.2.3	Extended sharing automation hierarchies	10
4.3	Two tier approach	11
4.4	Generic vs. user-controlled behavior	11
5	Gateway Architecture	12
5.1	General.....	12
5.2	Integration modules	13
5.2.1	General	13
5.2.2	Client side of an integration module	13
5.2.3	Gateway manager side of an integration module.....	13
5.2.4	ClientID handling	13
5.2.5	Reuse of existing integration specifications for integration modules	13
5.3	Access Rights Model.....	14
5.3.1	General	14
5.3.2	Service Categories	14
5.3.3	Access Rights.....	14
5.4	Client/interface coordination mechanism	15
5.4.1	General	15
5.4.2	Events	15
5.4.3	Concurrency and prioritization of SMI services	15
5.4.4	Exclusive Acyclic Access	15
5.4.5	Exclusive PDControl	17
5.5	The Standardized Gateway Interface "SGI"	19
5.5.1	General	19
5.5.2	SGI service arguments	19
5.5.3	SGI Services	20
5.5.4	SGI_GWM_Identification.....	21
5.5.5	SGI_GetClientNames	22
5.5.6	SGI_SetAccessRights.....	22
5.5.7	SGI_GetAccessRights	22
5.5.8	SGI_ExclusiveAcyclicAccess	22
5.5.9	SGI_SetGWMDData	22
5.5.10	SGI_GetGWMDData	22
5.5.11	SGI_SetPDControlPrio	22
5.5.12	SGI_GetPDControlPrio	22

5.5.13	SGL_ReleasePDControl	22
5.5.14	SGL_GetStatus	22
Annex A (normative) SMI and SGL service categories		23
A.1	Overview of service categories	23
Annex B (normative) Event Codes		24
B.1	Event Codes for Gateway management	24
B.2	Active exclusive acyclic access (0xFF00).....	24
B.3	Active exclusive PDOOut access (0xFF01).....	24
B.4	Change of exclusive PDOOut access (0xFF02)	24
Annex C (normative) Coding of ArgBlocks		25
C.1	Overview / Coding of ArgBlocks	25
C.2	GWM_Ident.....	25
C.3	GWM_ClientNames.....	25
C.4	GWM_AccessRights	26
C.5	GWM_AccessRightsResponse	27
C.6	GWM_ExclusiveAcyclicAccess.....	27
C.7	GWM_Data	28
C.8	GWM_PDControlPrio	28
C.9	GWM_Address.....	29
C.10	GWM_ReleasePDControl.....	29
C.11	GWM_Status.....	29
Annex D (normative) SGL ArgBlock ErrorTypes		31
D.1	SGL ArgBlock ErrorTypes	31
D.1.1	SGL associated ErrorTypes	31
D.1.2	CLIENT_ID_INVALID	31
D.1.3	MODULE_NUMBER_INVALID	31
D.1.4	PORT_NUM_INVALID	32
D.1.5	ARGBLOCK_NOT_SUPPORTED.....	32
D.1.6	ACCESS_NOT_GRANTED	32
D.1.7	ACCESS_TEMPORARILY_NOT_GRANTED.....	32
D.1.8	ARGBLOCK_LENGTH_INVALID.....	32
D.1.9	ARGBLOCK_INCONSISTENT	32
D.1.10	MEMORY_OVERRUN.....	32
D.1.11	GWM_ADDRESS_INVALID	32
D.1.12	ACCESS_RIGHTS_TOO_STRICT	32
Annex E (normative) Test strategies		33
E.1	General.....	33
E.2	Test ??	33
Bibliography.....		34

Figures

Figure 1 – Sample integration documents	9
Figure 2 – "Classic" fieldbus-based Automation Hierarchy	10
Figure 3 – Extended Automation Hierarchy with Gateway Management	11
Figure 4 – Vendor Level View of Gateway Architecture	12
Figure 5 – Integration modules	12
Figure 6 – Example for assignment of access rights.....	15
Figure 7 – Interface coordination via ClientIDs.....	15
Figure 8 – State machine of the Exclusive Access Manager.....	16
Figure 9 – State machine of the Exclusive PD manager	18
Figure 10 – SMI and SGI services with associated layers	21

Tables

Table 1 – Service categories.....	14
Table 2 – State transition tables of the Exclusive Access Manager	16
Table 3 – State transition tables of the Exclusive PD manager.....	18
Table 4 – SGI service request arguments	19
Table 5 – SGI service positive confirmation arguments	19
Table 6 – SGI service negative confirmation arguments.....	19
Table 7 – SGI services.....	21
Table A.1 – Category list of SMI and SGI services	23
Table B.1 – Event Codes for Gateway management	24
Table C.1 – ArgBlock types and their ArgBlockIDs	25
Table C.2 – ArgBlock GWM_Ident.....	25
Table C.3 – ArgBlock GWM_ClientNames.....	26
Table C.4 – ClientIDName	26
Table C.5 – ArgBlock GWM_AccessRights	26
Table C.6 – AccessRightGroup	26
Table C.7 – ArgBlock GWM_AccessRightsResponse	27
Table C.8 – AccessRightErrorInfo	27
Table C.9 – ArgBlock GWM_ExclusiveAcyclicAccess.....	27
Table C.10 – ArgBlock GWM_Data	28
Table C.11 – ArgBlock GWM_PDControlPrio	28
Table C.12 –PDControlPrioSet.....	28
Table C.13 – ArgBlock GWM_Address	29
Table C.14 – ArgBlock GWM_ReleasePDControl	29
Table C.15 – ArgBlock GWM_Status.....	29
Table D.1 – SGI ArgBlock related ErrorTypes	31
Table D.2 – Priority of ArgBlock ErrorTypes.....	31

1 Introduction

1.1 Scope

This document is for designers and implementers, who want to build a gateway between IO-Link Masters or modules with communication interfaces according to the SMI structure as defined in IO-Link Interface and System [1] on one side – and multiple integration technologies on the client side as higher level e.g. PLC / OPC UA / JSON technologies, etc.

This approach extends the classic PLC (classic process data processing) solutions that nowadays generally IoT and cloud applications require.

Therefore, the specification strives for a maximum of access separation and responsibility between different clients that may have simultaneously access to an underlined IO-Link system with clearly defined rules.

As a result, this specification establishes the opportunity for an intelligent gateway, managing the communication between the IO-Link Master and Client level, including the assignment of access rights via parameterization depending on the application requirements.

A basic target of this document is the achievement of a unified user experience for users of IO-Link Gateways. This includes a common precise understanding of access rights and basic functionality.

By a standardized Gateway Manager, the realization of generic configuration tools of the Gateway Manager (e. g. an extended PCDT) that fit different Gateway Manager brands, shall be simplified for further improvement of a unified user experience.

A further target is the definition of an abstract application programming interface, to make IO-Link Gateways easily accessible to the IT world and simplifies for example the design of reusable integration modules (see 3.1.2) which is not only advantageous for a unified user experience but also for the pervasiveness of IO-Link.

1.2 List of affected patents

There are no affected patents known by the members of the IO-Link gateway management working group. The list is empty. The IO-Link Community does not guarantee the completeness of this list.

2 References

The following document, in whole or in part, is referenced in this document and is indispensable for its application:

IO-Link Community, *IO-Link Interface and System*, V1.1.3, June 2019, Order No. 10.002 et seq.

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document apply

- the terms and definitions given in [1]
- the following specific definitions

3.1.1

Access Right Set

combination of low-level access rights that are linked to grant access to services

3.1.2

Integration module

gateway part providing the integration interface to be accessed by clients

3.1.3

Client

user application, which is accessing IO-Link data via an integration interface

3.1.4**ClientID**

element of an SMI / SGI service as defined in 11.2.2 in [1]

3.1.5**Gateway**

component representing a bridge between multiple communication sections

3.1.6**Gateway manager**

software agent using the implemented access rules parametrized and defined by SGI services, to control proper communication

3.1.7**Gateway Manufacturer**

entity providing Gateways with dedicated feature set for use by customers

3.1.8**ModuleNumber**

element of an SGI ArgBlock that addresses one SMI module

3.1.9**not applicable**

n/a

this entry cannot be applied within this context

3.1.10**SMI module**

unit providing services of the Standardized Master Interface

3.1.11**GWM-Address**

combination of ClientID or TargetClientID, ModuleNumber, PortNumber that specifies a set of one or more paths between a (Target)ClientID and a PortNumber of a specific SMI module.

3.2 Symbols and abbreviated terms

Device	IO-Link Device according [1]
GWM	Gateway Manager
JSON	Java Script Object Notation
Master	IO-Link Master according [1]
OPC UA	Open Platform Communications Unified Architecture
PDCT	Port and Device Configuration Tool ("Master Tool")
PLC	Programmable logic controller
Port	IO-Link Port according [1]

4 Involved Technologies**4.1 IO-Link technology**

The IO-Link technology is standardized within IEC 61131-9 as SDCI and supported by IO-Link Community documents (see [1]). It provides a digital communication for devices to exchange Process Data, diagnosis information and parameters with a controller (PC or PLC).

Communication is "point-to-point" from an IO-Link Master Port to an IO-Link Device. An IO-Link Master can have multiple ports and thus communicates simultaneously with multiple Devices.

IO-Link provides a standardized Master Interface (SMI) to ensure a harmonized behavior and communication over all Master manufactures into upper networks. This standardization was first released in V1.1.3 (see [1]).

4.2 Integration technologies

4.2.1 General

The increasing usage of IO-Link as a simple, powerful, and reliable interface to sensors and actuators has led to many proprietary and standardized integrations. The increasing usage of web access and IoT leads to applications managed by the user, referred to as integration technology. To optimize and support the integration technologies, IO-Link Community guidelines are used by all technology providers. E. g. referenced to fieldbus integration technologies (IEC 61158 and IEC 61784-1/2), see [2], [3] and [4].

The IO-Link Community themselves provides extended guideline definition for technology integrations.

- OPC UA technology (Companion standard for IO-Link), see [5]
- JSON mapping for IO-Link, see [6]

Some examples of integration documents are given in Figure 1.

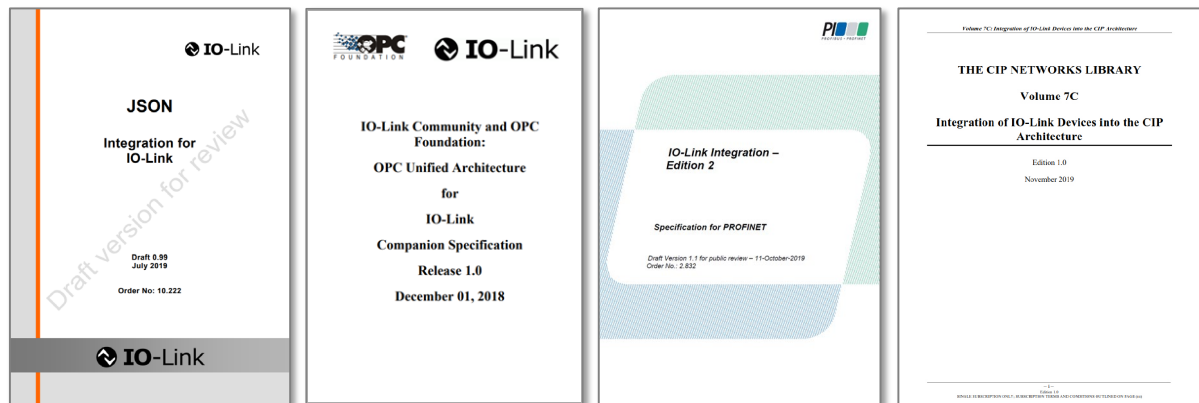


Figure 1 – Sample integration documents

IO-Link Gateways can be considered as agents between multiple client applications as PLC programs, asset management systems, predictive-maintenance-, data mining-, quality control-, or Event logger software that need simultaneous access to IO-Link Devices connected to IO-Link Masters.

Authentication and security aspects for the Gateway are handled by integration technology and out of scope of the Gateway Manager.

4.2.2 Classic fieldbus-based automation hierarchy

Within the classic fieldbus-based automation hierarchy according to Figure 2, all Devices are controlled exactly by one client – usually by the fieldbus application. This fieldbus application essentially has unrestricted access to the functionality of the Device. Concurrent accesses are handled within the gateway (fieldbus).

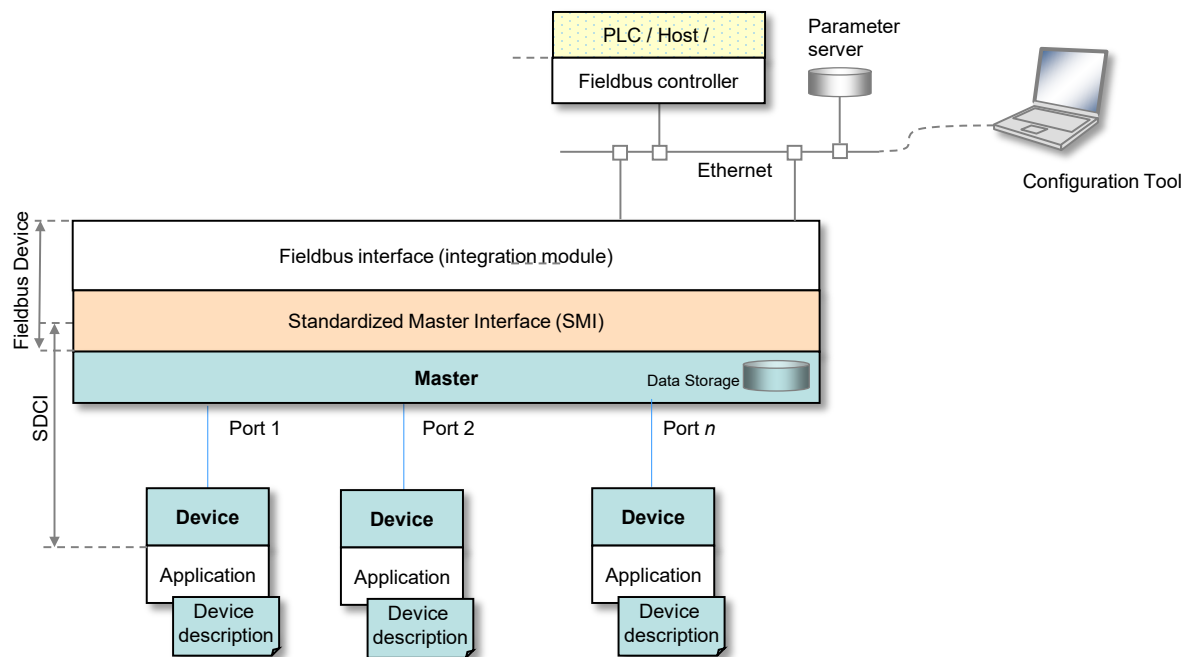


Figure 2 – "Classic" fieldbus-based Automation Hierarchy

4.2.3 Extended sharing automation hierarchies

The standard master interface SMI introduced in version 1.1.3 of the IO-Link System and Interface specification [1] offers a well-defined access point to different client applications.

Simultaneous access of different clients to a single IO-Link component, however, may lead to undesired or even fatal consequences for the automation system.

For example, if a PLC writes process data to a system that causes a switch to close and simultaneously an internet-based client sends process data to open the switch, the behavior of the system may become unpredictable.

The situation described in this example can be avoided by a policy that only one client has permission to write process data.

The gateway management provides means to implement and enforce policies that refer to potential issues caused by simultaneous access of clients to IO-Link components.

Such means are for example:

- Assignment of different levels for different clients.
- Registration of notifications indicating configuration changes.

While the definition and description of the means to avoid potential issues are within the scope of this document, the policies are not.

In fact, they may be completely different as in the case of automation systems. The definition of policies is thus under the responsibility of the system designer.

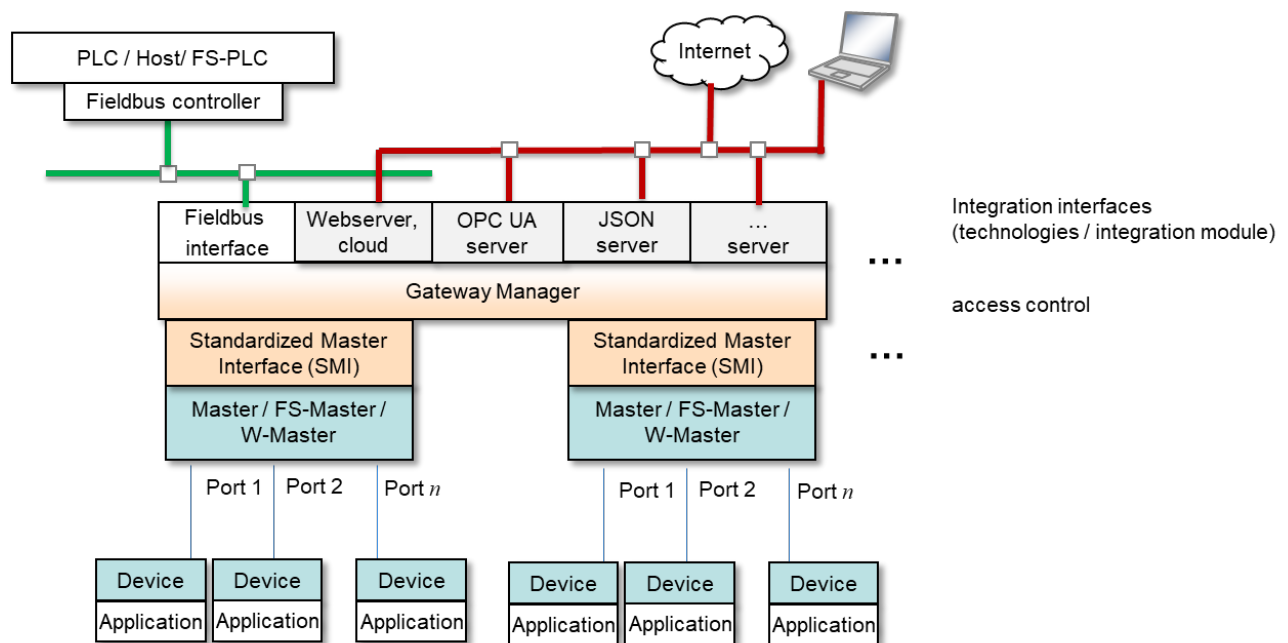


Figure 3 – Extended Automation Hierarchy with Gateway Management

Figure 3 shows an example of an automation system with multiple clients and multiple SMI modules e.g. IO-Link Masters.

However, to support the use of GWM Configuration tools for different Gateway manager brands a harmonized interface called SGI with explicit identification of the supported gateway modules is defined in this document.

4.3 Two tier approach

Objective of this document is creating a uniform method for the coordination of different integration interfaces. This has the two aspects of

- how this coordination feature is presented to the customers/users, and
- which kind of algorithms and software technologies should be implemented.

A two-tier approach shall lead to a final specification. The first tier is heading for an agreed upon user view and the second tier will comprise implementation recommendations.

It is not an objective of this document to coordinate different clients within one integration technology/interface. For example, the question of how to deal with different clients using the OPC UA server is the responsibility of this integration technology document.

4.4 Generic vs. user-controlled behavior

Two principles for coordination are possible and for the IO-Link gateway it has not been decided, which to follow.

It would always be the most convenient way for customers/users, if coordination could be achieved automatically via a generic behavior. However, since the gateway cannot acquire information on how clients are using the integration interfaces, a generic behavior is difficult to design, and the method would be most likely error prone.

The other method provides a configuration tool to the user for the assignment of authorizations/prioritizations to achieve a stable system behavior.

5 Gateway Architecture

5.1 General

Examples for user level views of a basic access rights assignment are given below.

The vendor level view of the gateway architecture is depicted in Figure 4. It shows the vendor level view of the architectural structure of an IO-Link gateway. The vendor level view of the gateway describes the data flow hierarchy from user access points provided by integration technologies down to the IO-Link Devices at IO-Link Master Ports.

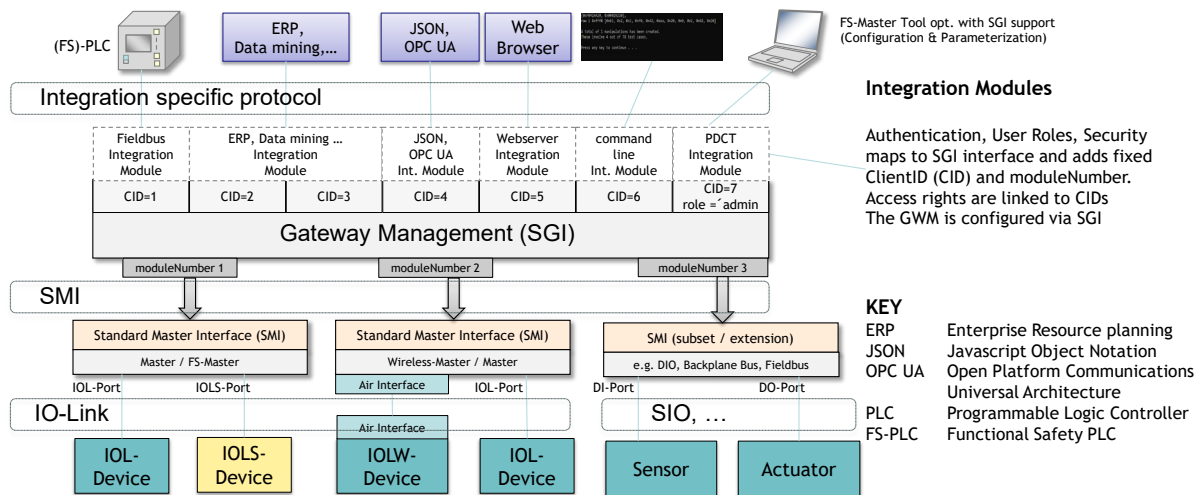


Figure 4 – Vendor Level View of Gateway Architecture

The elements of the Gateway Architecture form a vendor level viewpoint in Figure 4 are:

- Integration modules – protocol conversion (see 5.2).
- Gateway Manager GWM – managing access rights and additional functionality (see 5.3).
- SMI Modules – simultaneously accessible (see 13 in [1]).

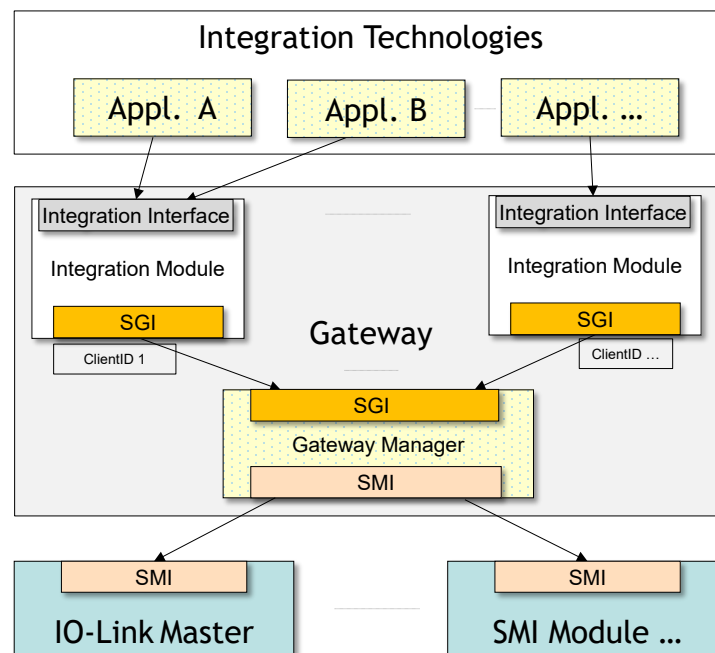


Figure 5 – Integration modules

As shown in Figure 5, integration modules are an integral part of the gateway.

5.2 Integration modules

5.2.1 General

Integration modules realize the conversion of elements, e.g. services, objects, or events of the integration technology into a common format: the SGI interface (see 5.5).

The gateway manufacturer specifies, which integrations are supported by the gateway and how the abstract interfaces SGI and SMI are realized. As shown in Figure 5, the integration modules have a client-side end point that interact with one or more communication elements of the integration technology. They also have a gateway-manager-side that interacts with the gateway manager.

5.2.2 Client side of an integration module

The different client applications (Fieldbus, OPC UA, ...) have their own integration technology. A specific protocol related to the integration technology defines the connection to the client side, showing access points, and the connection to the Gateway manager side, showing by unique ClientIDs. The client-side of an integration module may have one or more interfaces that allow different applications of the corresponding integration technology to connect to the corresponding integration module. Typical tasks of the client-side integration modules are:

- Client / User authentication
- Client / User role identification
- Object Mapping
- Concurrency inside the same type (e.g. access of more than one OPC UA client)
- Security Encryption – interception protection
- Routing of applications based on the integration technology to ClientIDs.

5.2.3 Gateway manager side of an integration module

The gateway-manager-side of an integration module uses the ClientID of the SGI header as discriminator for assigning access rights. Instead of regular SMI services, SGI services are used for communication, which basically are SMI services extended with a ModuleNumber, see 5.5.2. The structure of SGI Services is described in 5.5.

5.2.4 ClientID handling

The Gateway manufacturer assigns ClientIDs to the integration module. The ClientID has two functions within the Gateway Management:

- Client identification for access right assignment.
- Routing information to match request and asynchronous response part of an SGI service.

The fact that access rights are linked to ClientIDs within an SMI service requires that the assignment of the ClientID stays under control of the designer/manufacturer of the Gateway.

ClientIDs must not be selectable by the integration technology, in fact they even do not need to be visible in the user level view.

In configuration tools, the ClientName is used as alias for the ClientID. The ClientName shall indicate the integration technology and possible instances with a unique user friendly name like "RestAPI-Admin".

5.2.5 Reuse of existing integration specifications for integration modules

The gateway manufacturer can reuse existing integration modules that connect directly with an IO-Link Master via SMI (see Figure 2) by adding support for the ModuleNumbers to get SGI services. In its simplest form, this can be realized by a static assignment of the ModuleNumber.

The amount of supported SGI services depends on the necessary functionality and may differ between implementations.

In case an integration module is able to write PDOOut, the service SGI_ReleasePDControl shall be supported.

In case of missing access rights at the GWM, a service shall be blocked (receive a JobError response with ErrorType ACCESS_NOT_GRANTED, see D.1.6).

In case of temporarily blocked single resources like process data or acyclic access, a service shall be blocked (receive a JobError response with ErrorType ACCESS_TEMPORARILY_NOT_GRANTED, see D.1.7).

The integration module shall handle the additional error codes.

5.3 Access Rights Model

5.3.1 General

For reasons of simplicity, individual access rights (for example, the right to write an IO-Link Parameter) cannot be assigned directly, instead access rights are assigned separately for read and write to a set of categories of services (see 5.3.2). The access right for one of the service categories for a ClientID can be assigned individually by a combination of ModuleNumber and PortNumber (see 5.3.3).

Access rights are pre-configured by the gateway manufacturer and can be modified via the GWM Services (see 5.5).

5.3.2 Service Categories

The SMI services and the SGI services, are grouped into five service categories given in Table 1.

Table 1 – Service categories

Service category	Description
System data	Global configuration of GWM like access rights (SGI services)
Module / Port data	Port specific configuration depending on technology (Master services)
Device data	Configuration of device for application (Device services)
Process data	Process data access in general
Unrestricted	Unconstrained services, not part of access right restrictions

For a detailed list of services, see Table A.1.

5.3.3 Access Rights

The access rights specify the actions a client may perform interacting with the gateway manager, SMI modules, IO-Link Masters, Master-Ports, or connected IO-Link Devices.

There are three different access rights which can be configured:

- No Access: inhibits any access to a service of the given service category,
- Read Only (R): allows only read services of the given service category to be executed,
- Read / Write (R/W): allows all services of the given category to be executed.

Rights for a service category are assigned for a combination of ClientID, ModuleNumber and PortNumber. To avoid excessive length of the rights at commissioning, wildcards can be used for ClientID, ModuleNumber, and PortNumber.

Figure 6 shows an exemplary assignment of access rights for one port of a Master with several clients.

Access rights example:

Client name	Profinet	Ethernet/IP	Web	OPC UA	REST/JSON	...
ClientID	#1	#2	#3	#4	#5	... #n
System data	R ☒ W ☐	R ☒ W ☐	R ☒ W ☒	R ☐ W ☐	R ☐ W ☐	R ☐ W ☐
Module / Port data	R ☒ W ☒	R ☒ W ☒	R ☒ W ☐	R ☐ W ☐	R ☐ W ☐	R ☐ W ☐
Device data	R ☒ W ☒	R ☒ W ☒	R ☒ W ☐	R ☒ W ☐	R ☒ W ☐	R ☐ W ☐
Process data	R ☒ W ☒	R ☒ W ☒	R ☒ W ☒	R ☒ W ☐	R ☒ W ☐	R ☒ W ☐
Process data write priority	1 (high)	1 (high)	3 (low)	255	255	255

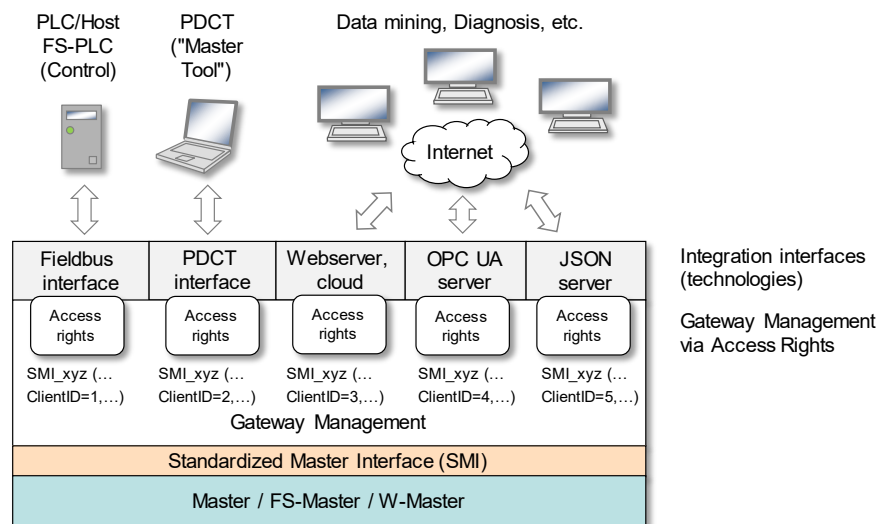
Figure 6 – Example for assignment of access rights

For more complex embodiments the assign access rights for each module and/or port can be set individually.

5.4 Client/interface coordination mechanism**5.4.1 General**

The gateway must ensure the SMI services are performed conflict-free to guarantee a reliable automation system.

The SMI specification in [1] supports a client coordination mechanism using a unique ClientID handle in each SMI service invocation for responding to the correct invoking client (see Figure 7).

**Figure 7 – Interface coordination via ClientIDs****5.4.2 Events**

The Gateway management is responsible to forward the push services SMI_DeviceEvent and SMI_PortEvent as well as SGI events to all interfaces. If the GWM issues an Event, the INSTANCE field shall be set to "4" (Application), see Table A.17 in [1].

5.4.3 Concurrency and prioritization of SMI services

This concept is specified in clause 11.2.3 of [1]. Prioritization of SGI and SMI services are not performed. All services accessing methods will be processed in the order of their arrival (first come, first serve).

5.4.4 Exclusive Acyclic Access

Figure 8 defines the different states how to request singular access and block concurrent accesses by other clients meanwhile to provide undisturbed and consistent access.

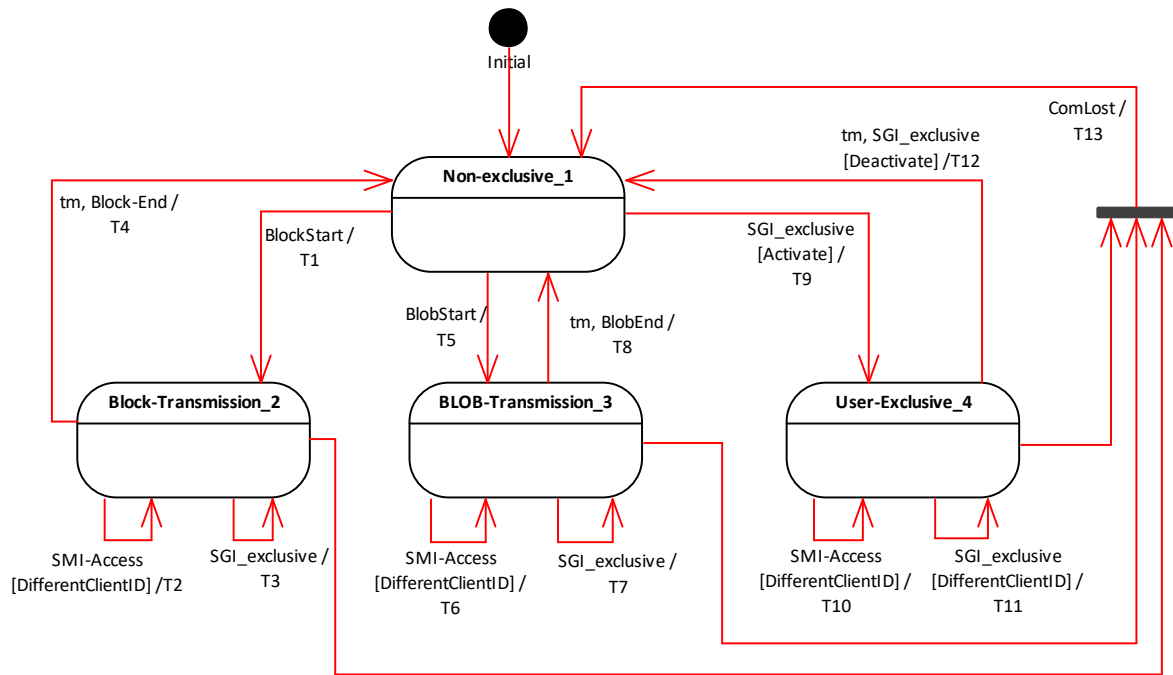


Figure 8 – State machine of the Exclusive Access Manager

Table 2 shows the state transition tables of the Exclusive Access Manager.

Table 2 – State transition tables of the Exclusive Access Manager

STATE NAME	STATE DESCRIPTION		
Non_Exclusive_1	There is no exclusive acyclic access ongoing.		
Block-Transmission_2	A block transmission is ongoing by the ActiveClientID. Acyclic accesses are granted for the ActiveClientID and each access resets the timer tm.		
BLOB-Transmission_3	A BLOB transmission is ongoing by the ActiveClientID. Acyclic accesses are granted for the ActiveClientID and each access resets the timer tm.		
User-Exclusive_4	A Client explicitly requested exclusive acyclic access. Acyclic accesses are granted for the ActiveClientID and each access resets the timer tm.		

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	1	2	Set ActiveClientID to ClientID of requester. Broadcast appearing Event 0xFF00. Set AcycExclusiveCause of GWM_Status to "Block transfer".
T3	2	2	Return ErrorType 0x4144
T4	2	2	Return ErrorType 0x4144.
T5	2	1	Broadcast disappearing Event 0xFF00. Set ActiveClientID to "NONE". Set AcycExclusiveCause of GWM_Status to "No exclusive access".
T6	1	3	Set ActiveClientID to ClientID of requester. Broadcast appearing Event 0xFF00. Set AcycExclusiveCause of GWM_Status to "BLOB transfer".
T8	3	3	Return ErrorType 0x4144
T9	3	3	Return ErrorType 0x4144
T10	3	1	Broadcast disappearing Event 0xFF00. Set ActiveClientID to "NONE". Set AcycExclusiveCause of GWM_Status to "No exclusive access".
T11	1	4	Set ActiveClientID to ClientID of requester. Broadcast appearing Event 0xFF00. Set AcycExclusiveCause of GWM_Status to "User request".
T13	4	4	Return ErrorType 0x4144
T14	4	4	Return ErrorType 0x4144

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T15	4	1	Broadcast disappearing Event 0xFF00. Set ActiveClientID to "NONE". Set AcycExclusiveCause of GWM_Status to "No exclusive access".
T16	2, 3, 4	1	Broadcast disappearing Event 0xFF00. Set ActiveClientID to "NONE". Set AcycExclusiveCause of GWM_Status to "No exclusive access".
INTERNAL ITEMS		TYPE	DEFINITION
BlockStart		Trigger	SystemCommand "ParamDownloadStart" or "ParamUploadStart" is successfully written by a client, see [1] for details.
BlockEnd		Trigger	Successful SystemCommand "ParamUploadEnd", "ParamDownloadEnd", "ParamBreak", "ParamDownloadStore", any of the reset SystemCommands (0x80 – 0x83), or any other condition (e. g. PortStatusChanged, ...) detectable by the GWM, which terminates the BlockTransfer (see Device ParameterManager state machine in 10.3.2 in [1] for details).
BLOBStart		Trigger	A BLOB transfer is successfully started by a client, see [7] for details.
BLOBEnd		Trigger	The BLOB transfer has been completed or otherwise terminated, see [7] for details.
SGI_exclusive		Call	Successful invocation of SGI_ExclusiveAcyclicAccess.
Activate		Bool	RequestAccess = true, see C.6.
Deactivate		Bool	RequestAccess = false, see C.6.
SameClientID		Bool	ClientID of requester is equal to ActiveClientID
DifferentClientID		Bool	ClientID of requester is not equal to ActiveClientID
ComLost		Trigger	A "No Device" Port Event was received from the corresponding port.
tm		Timer	AutomaticTimeOut, see C.6
ActiveClientID		variable	Stores the ID of the client, which currently has exclusive access to parameters. "None" represents that no client is currently active.

5.4.5 Exclusive PDControl

Write access to process data is more restricted than to other data categories: At one point in time only one client shall have write access to process data out. This is guaranteed by the process data write priority. Process data write permission shall be granted to the client with the highest process data control priority. A process data write service of a client with lower priority shall be refused then. A process data writes service of a client with a priority equal to the priority of a client that has already process data write permission shall also be refused (first come, first served). The integration is responsible for release of the PDOut control right by issuing the SGI_ReleasePDControl service in case of connection lost or timeout of the integration technology.

Figure 9 defines the different states how to handle the exclusive process data access based on the priorities and Table 3 shows the state transition tables of the Exclusive PD Manager.

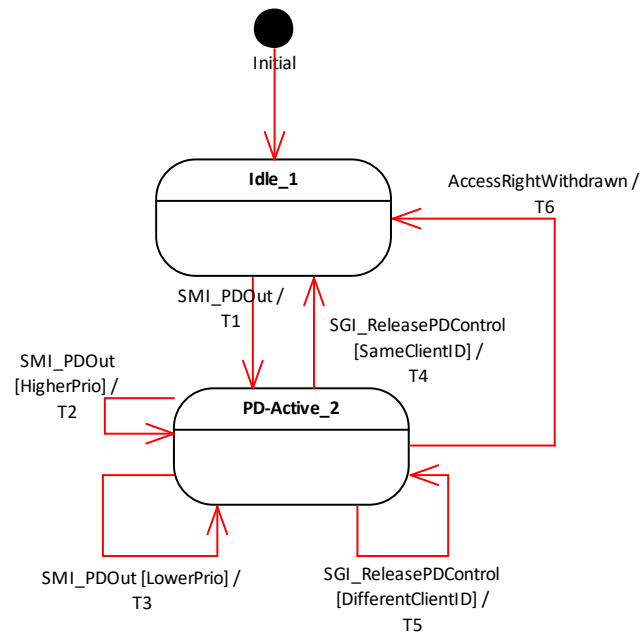


Figure 9 – State machine of the Exclusive PD manager

Table 3 – State transition tables of the Exclusive PD manager

STATE NAME		STATE DESCRIPTION	
Idle_1		No control of PDOOut and PDOOut set to invalid.	
PD_Active_2		Control of PDOOut by ActiveClientID with controlled validity.	
TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	1	2	Set ActiveClientID to ClientID of requester. Set ActivePDControlPrio to PDControlPrio of requester. Set PDOOut according request. Broadcast appearing Event 0xFF01.
T2	2	2	Set ActiveClientID to ClientID of requester. Set ActivePDControlPrio to PDControlPrio of requester. Set PDOOut according request. Broadcast access change by Event 0xFF02.
T3	2	2	Negative confirmation of SMI_PDOut service with ErrorType 0x4143.
T4	2	1	Set ActiveClientID to "NONE". PDOOut and PDValidity shall be unchanged. Broadcast disappearing Event 0xFF01.
T5	2	2	Negative confirmation of SGI_ReleasePDControl ErrorType 0x4142.
T6	2	1	Broadcast disappearing Event 0xFF01.
INTERNAL ITEMS		TYPE	DEFINITION
PDControlPrio		Data	PDControlPrio of the requesting client
ActiveClientID		Variable	Stores the client identification with current exclusive access to process data. "None" represents that no client is currently active.
ActivePDControlPrio		Variable	Stores the PDControlPrio of the ActiveClientID
HigherPrio		Bool	Active when PDControlPrio is higher than ActivePDControlPrio
LowerPrio		Bool	Active when PDControlPrio is equal or lower than ActivePDControlPrio

INTERNAL ITEMS	TYPE	DEFINITION
SameClientID	Bool	ClientID of requester is equal to ActiveClientID
DifferentClientID	Bool	ClientID of requester is not equal to ActiveClientID
AccessRightWithdrawn	Service	Activated when the access rights are changed via access rights changes or ComLost

5.5 The Standardized Gateway Interface "SGI"

5.5.1 General

SGI services are structured like SMI services defined in [1]. For addressing multiple SMI modules, an additional argument ModuleNumber is added to SMI and SGI services.

An SGI supports the following types of services:

- The SGI services listed in Table 7 are interacting with the GWM itself.
- SMI services are propagated to a specific SMI module addressed by the service argument ModuleNumber.

5.5.2 SGI service arguments

The SGI services contain a fixed structure of standard elements defined in the following tables.

Table 4 – SGI service request arguments

Parameter name	.req	.cnf
ModuleNumber	M	
ClientID	M	
PortNumber	M	
ExpArgBlockID	M	
ArgBlockLength	M	
ArgBlock	M	

Table 5 – SGI service positive confirmation arguments

Parameter name	.req	.cnf
ModuleNumber		M
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock)		M
ArgBlockLength		M
ArgBlock (associated to ExpArgBlockID)		M

Table 6 – SGI service negative confirmation arguments

Parameter name	.req	.cnf
ModuleNumber		M
ClientID		M
PortNumber		M
RefArgBlockID (ID of request ArgBlock)		M
ArgBlockLength		M
ArgBlock (JobError: 0xFFFF)		M

ModuleNumber

The number of the SMI module that is addressed by the SGI service. The ModuleNumber extends the SMI service arguments. SMI can address only one IO-Link Master, while SGI can address multiple SMI modules.

324 Data type: Unsigned8

325 Permitted values:

- 326 • "0" reserved for SGI services only (see Table 7).
- 327 • 1 .. n, with n the maximum number of concurrent modules

328

329 **ClientID**

330 The ClientID identifies a client and is unique within the gateway.

331 Data type: Unsigned8

332 Permitted values:

- 333 • "0" for broadcast purposes in case of Events.
- 334 • 1 .. n, with n the maximum number of concurrent clients.

335

336 **PortNumber**

337 Each SMI service contains the port number of the SMI module in case of an addressed port object
338 (job) or in case of a triggered port object (event).

339 Data type: Unsigned8

340 Permitted values:

- 341 • "0" reserved for SGI services only (see Table 7).
- 342 • 1 .. n, with n the maximum number of ports.

343

344 **ExpArgBlockID/RefArgBlockID**

345 This element specifies the expected/referred ArgBlockID to carry the response data of a service
346 request.

347 Data type: Unsigned16

348 Permitted values: 1 to 65535

349

350 **ArgBlockLength**

351 This element specifies the total length of the subsequent ArgBlock. Vendor specific extensions are
352 not permitted.

353 Data type: Unsigned16

354 Permitted values: 2 to 65535

355

356 **ArgBlock**

357 All SMI services contain an ArgBlock characterized by an ArgBlockID and its description.

358 Service results provide the ArgBlock associated to the ExpArgBlockID, which is part of this
359 ArgBlock. The possibly variable length of the ArgBlock is predefined through definition in this
360 document.

361 **5.5.3 SGI Services**

362 The targets of the several services are shown in Figure 10 to highlight the differences between
363 SMI and SGI services and the point of access control. The SMI services in Figure 10 are only
364 exemplary and not complete as several SMI services are defined in various specifications.

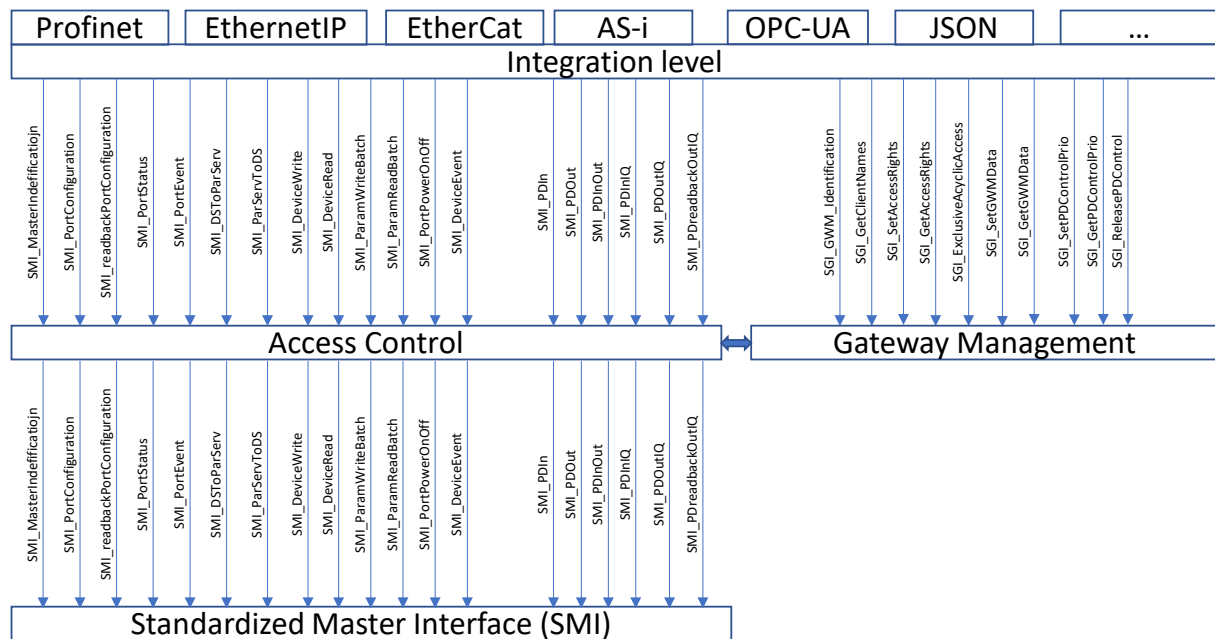


Figure 10 – SMI and SGI services with associated layers

The SGI services shown in Figure 10 are listed in Table 7 below. All services are mandatory, the GWM itself is always the Receiver/Responder of the service.

In case of an error, all services return a JobError Argblock, although not explicitly mentioned for each service in the CNF ArgBlock column in **Error! Reference source not found..** Permitted values for ErrorType in JobError in prioritized order see Table D.2, possible associations are defined in the ErrorType description.

Table 7 – SGI services

Service name	REQ ArgBlock	CNF ArgBlock	Definition
SGI_GWM_Identification	VoidBlock	GWM_Ident (see C.2)	5.5.4
SGI_GWM_GetClientNames	VoidBlock	GWM_ClientNames (see C.3)	5.5.5
SGI_SetAccessRights	GWM_AccessRights (see C.4)	GWM_AccessRightsResponse (see C.5)	5.5.6
SGI_GetAccessRights	GWM_Address (see C.9)	GWM_AccessRights (see C.4)	5.5.7
SGI_ExclusiveAcyclicAccess SGI_ExclusiveAcyclicAccess	GWM_ExclusiveAcyclicAccess (see C.6)	VoidBlock	5.5.8
SGI_SetGWMDData	GWM_Data (see C.7)	VoidBlock	5.5.9
SGI_GetGWMDData	VoidBlock	GWM_Data (see C.7)	5.5.10
SGI_SetPDControlPrio	GWM_PDControlPrio (see C.8)	VoidBlock	5.5.11
SGI_GetPDControlPrio	GWM_Address (see C.9)	GWM_PDControlPrio (see C.8)	5.5.12
SGI_ReleasePDControl	GWM_ReleasePDControl (see C.10)	VoidBlock	5.5.13
SGI_GetStatus	VoidBlock	GWM_Status (see C.11)	5.5.14

5.5.4 SGI_GWM_Identification

This service returns identification of the Gateway Manager by the GWM_Ident ArgBlock. It allows GWM Configuration Tools to adjust to individual Gateway Manager brands and types.

5.5.5 SGI_GetClientNames

This service returns the human-readable names for all ClientIDs. It can be used by GWM Configuration Tools to show human-readable names in the UI instead of raw ClientIDs.

5.5.6 SGI_SetAccessRights

This service sets the access rights for individual ClientIDs for modules and ports. The types of access rights are defined in 5.3.

If one of the access rights cannot be set, the whole access right setting shall be rejected with the appropriate ErrorType.

It is on behalf of the gateway management implementation to ensure at least one client with sufficient access to handle the gateway management settings, violations shall be responded by ErrorType ACCESS_RIGHTS_TOO_STRICT.

5.5.7 SGI_GetAccessRights

This service returns the current access rights for GWM Addresses.

5.5.8 SGI_ExclusiveAcyclicAccess

This service requests or releases exclusive acyclic access by a specific client on a specific module and port to inhibit interfering acyclic accesses by other clients on this port for a short period of time. The dynamic behavior of the exclusive acyclic access is specified in 5.4.4.

Exclusive access can only be requested by the requesting client identified by ModuleNumber, PortNumber, and ClientID which provided by the service arguments.

5.5.9 SGI_SetGWMDData

The service SGI_SetGWMDData can be used to restore the current configuration of the GWM from a prior data backup via SGI_GetGWMDData.

The content of the GWM_Data is implementation dependent, the data usability is only guaranteed for GWMs of the same brand and compatible type. It is on behalf of the GWM to check the consistence, integrity, and compatibility to prevent misbehavior.

5.5.10 SGI_GetGWMDData

The service SGI_GetGWMDData returns the current configuration of the GWM for backup purposes and restoring the GWM itself. The data is not defined in structure or content and is defined by the manufacturer of the GWM.

5.5.11 SGI_SetPDControlPrio

This service sets the priority of PDOOut write access on a module and port for a clients. The priority is used for grant process data out setting only for the client with the highest priority according to 5.4.5.

5.5.12 SGI_GetPDControlPrio

This service returns the priority of PDOOut write access settings for GWM Addresses.

5.5.13 SGI_ReleasePDControl

This service requests to release the write access right by the calling client.

5.5.14 SGI_GetStatus

This service provides the current state of restricted accesses to acyclic data and process data.

Annex A (normative)

SMI and SGI service categories

A.1 Overview of service categories

As specified in 5.3, the access rights can be defined categories of services of common functionality. The specified services in [1] and in this document are listed in Table A.1. Extensions providing similar services shall fit these into the most appropriate category.

Table A.1 – Category list of SMI and SGI services

Service Category	Services	Category description
System data	SGI_SetAccessRights	Global configuration of GWM like access rights or commissioning
	SGI_GetAccessRights	
	SGI_SetGWMDData	
	SGI_GetGWMDData	
	SGI_SetPDControlPrio	
	SGI_GetPDControlPrio	
Module / Port data	SMI_PortStatus	Port specific configuration depending on technology targeting the IO-Link Master itself
	SMI_PortPowerOffOn	
	SMI_PortConfiguration	
	SMI_ReadbackPortConfiguration	
Device data	SMI_DSToParServ	Configuration of Device for application targeting the IO-Link Device itself
	SMI_ParServToDS	
	SMI_DeviceWrite	
	SMI_DeviceRead	
	SMI_ParamWriteBatch	
	SMI_ParamReadBatch	
	SGI_ExclusiveAcyclicAccess	
Process data	SMI_PDIn	Runtime access to process data
	SMI_PDOut	
	SMI_PDInOut	
	SMI_PDInIQ	
	SMI_PDOutIQ	
	SMI_PDReadbackOutIQ	
	SGI_ReleasePDControl	
Unrestricted	SMI_DeviceEvent	These types of services are unconstrained and cannot be restricted in access
	SMI_PortEvent	
	SGI_GWM_Identification	
	SMI_MasterIdentification	
	SGI_GetClientNames	
	SGI_GetStatus	

Annex B (normative)

Event Codes

B.1 Event Codes for Gateway management

Table B.1 lists the specified EventCode identifiers and their definitions for Gateway management (SOURCE = "Master/Port").

Table B.1 – Event Codes for Gateway management

EventCode ID	Definition and recommended maintenance action	Event Instance	Type
0xFF00	Signaling the access restriction state by activating exclusive acyclic access of one client	APP	Warning
0xFF01	Signaling the access restriction state by activating the exclusive PDOOut access by one client	APP	Warning
0xFF02	Signaling the change of PDOOut access	APP	Notification

B.2 Active exclusive acyclic access (0xFF00)

The state of this event represents the access restriction state for acyclic accesses at one specific Module/Port by a specific client. Any acyclic access by other clients will be rejected during this phase, see Figure 8.

B.3 Active exclusive PDOOut access (0xFF01)

The state of this event represents the access restriction state for PDOOut at one specific Module/Port by a specific client. Any PDOOut access by other clients with equal or lower priority will be rejected during this phase, see Figure 9.

B.4 Change of exclusive PDOOut access (0xFF02)

This event notifies the clients on changes of the ownership of exclusive PDOOut access, see Figure 9.

Annex C (normative)

Coding of ArgBlocks

C.1 Overview / Coding of ArgBlocks

Coding of ArgBlockIDs is specified in Figure E.1 in [1]. The gateway management is identified by the domain identifier 0xD.

Table C.1 provides a list of all SGI ArgBlocks defined in this document.

Table C.1 – ArgBlock types and their ArgBlockIDs

ArgBlock type	ArgBlock ID	Description	Definition
GWM_Ident	0x0D01	Identification of GWM	C.2
GWM_ClientNames	0x0D02	Mapping of ClientIDs to names	C.3
GWM_AccessRights	0x7D03	Specify access rights	C.4
GWM_AccessRightsResponse	0x7D04	Response to setting access rights	C.5
GWM_ExclusiveAcyclicAccess	0x7D05	Exclusive access	C.6
GWM_Data	0x7D06	Image of SGI management configuration	C.7
GWM_PDControlPrio	0x7D07	Priority information for PDOOut control management	C.8
GWM_Address	0x7D08	Used for reading AccessRights and PDControlPrios	C.9
GWM_ReleasePDControl	0x7D09	Used to indicate that a clients stops writing PDOOut for the given module/port	C.10
GWM_Status	0x7D0A	Provides current status of exclusive access paths	C.11

C.2 GWM_Ident

The ArgBlock GWM_Ident defined in Table is used by the service SGI_GWM_Identification specified in 5.5.4.

Table C.2 – ArgBlock GWM_Ident

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0D01
2	VendorID	Unique VendorID of Gateway Manager (see [1, B.1.8])	Unsigned16	0x01 to 0xFFFF
4	GatewayManagerID	Unique vendor specific type identification of the Gateway Manager.	Unsigned32	0x01 to 0xFFFFFFFF
8	SGI version	highest SGI Version supported by the GWM. (Higher numbers refer to newer SGI Versions).	Unsigned8	0x01
9	Features_1	Reserved.	Unsigned8	0x00
10	Features_2	Reserved.	Unsigned8	0x00
11	SW_Version	Software version identifier of the Gateway Manager	StringT[32]	n/a

The length of this ArgBlock is 43 octets.

C.3 GWM_ClientNames

The ArgBlock GWM_ClientNames defined in Table C.3 is used by the service SGI_GetClientNames specified in 5.5.5.

Table C.3 – ArgBlock GWM_ClientNames

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x0D02
2	Count	This element provides the number N of following ClientIDNames	Unsigned8	0x01 to 0xFF
3	ClientIDNamesList	List of ClientIDNames, as defined in Table C.4.	Array[N] of ClientIDName	n/a

The length of this ArgBlock is 3 octets plus the length of the ClientIDName list.

Table C.4 – ClientIDName

Octet Offset	Element name	Definition	Data type	Values
0	TargetClientID	ClientID	Unsigned8	0x01 to 0xFF
1	Name	Human-readable name of the client identified by TargetClientID	StringT[32]	n/a

The length of a ClientIDName is 33 octets.

C.4 GWM_AccessRights

The ArgBlock GWM_AccessRights defined in Table C.5 is used by the services SGI_SetAccessRights and SGI_GetAccessRights specified in 5.5.6 and 5.5.7.

Table C.5 – ArgBlock GWM_AccessRights

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D03
2	Count	This element provides the number N of following access right groups	Unsigned8	0x01 to 0xFF
3	AccessRightGroupList	List of AccessRightGroups, as defined in Table C.6	Array[N] of AccessRightGroup	n/a

The length of this ArgBlock is 3 octets plus the length of the AccessRightsGroupList.

Table C.6 – AccessRightGroup

Octet Offset	Element name	Definition	Data type	Values
0	TargetClientID	ClientID associated to the rights. 0: all clients (only for SGI_SetAccessRights) 1 to 255: specific client	Unsigned8	0x00 to 0xFF
1	TargetModuleNumber	ModuleNumber associated to the rights. 0: all modules (only for SGI_SetAccessRights) 1 to 255: specific module	Unsigned8	0x00 to 0xFF
2	TargetPortNumber	PortNumber associated to the rights. 0: all ports (only for SGI_SetAccessRights) 1 to 255: specific port	Unsigned8	0x00 to 0xFF

Octet Offset	Element name	Definition	Data type	Values
3	AccessGroup	0: reserved 1: System data 2: Module/Port data 3: Device data 4: Process data 5 to 255: reserved	Unsigned8	0x01 to 0x04
4	Right	0: No Access 1: Read Only 2: Read / Write 3 to 255: reserved	Unsigned8	0x00 to 0x02

The length of an AccessRightsGroup is 5 octets.

C.5 GWM_AccessRightsResponse

The ArgBlock GWM_AccessRightsResponse defined in Table C.7 is used by the service SGI_GetAccessRights specified in 5.5.7.

Table C.7 – ArgBlock GWM_AccessRightsResponse

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D04
2	Count	This element provides the number N of following AccessRightErrorInfos	Unsigned8	0x01 to 0xFF
3	AccessRightErrorInfoList	List of AccessRightErrorInfo, as defined in Table C.8	Array[N] of AccessRightErrorInfo	n/a

The length of this ArgBlock is 4 octets plus the length of the AccessRightErrorInfoList.

Table C.8 – AccessRightErrorInfo

Octet Offset	Element name	Definition	Data type	Values
0	Index	Array index of GWM_AccessRights ArgBlock	Unsigned8	0x01 to 0xFF
1	ErrorCode	Detailed error information associated to the Index	Unsigned8	0: invalid TargetClientID 1: invalid TargetModuleNumber 2: invalid TargetPortNumber 3: invalid AccessGroup 4: invalid Right

The length of an AccessRightsErrorInfo is 2 octets.

C.6 GWM_ExclusiveAcyclicAccess

The ArgBlock GWM_ExclusiveAcyclicAccess defined in Table C.9 is used by the service SGI_ExclusiveAcyclicAccess specified in 5.5.8.

Table C.9 – ArgBlock GWM_ExclusiveAcyclicAccess

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D05
2	TargetModuleNumber	ModuleNumber	Unsigned8	0x01 to 0xFF
3	TargetPortNumber	PortNumber	Unsigned8	0x01 to 0xFF
4	RequestAccess	0: request acyclic access 1: release acyclic access	Unsigned8	0x00 to 0x01
5	AutomaticTimeOut	Timeout for automatic release in seconds 0: 10 seconds 10 to 255: 10 to 255 seconds	Unsigned8	0x00 and 0x0A to 0xFF

The length of this ArgBlock is 6 octets.

C.7 GWM_Data

The ArgBlock GWM_Data defined in Table C.10 is used by the services SGI_GetGWMDData and SGI_SetGWMDData specified in 5.5.10 and 5.5.9.

The size of the GWM_Data is indirectly defined by the ArgBlockLength field in the service arguments.

Table C.10 – ArgBlock GWM_Data

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D06
2	Data_Ident	Identification string for configuration data	StringT[32]	n/a
34	Data	Configuration data	Octet string	n/a

The length of this ArgBlock is 3 octets plus the length of the Data element.

C.8 GWM_PDControlPrio

The ArgBlock GWM_PDControlPrioSet defined in Table C.11 is used by the services SGI_SetPDControlPrio and SGI_GetPDControlPrio specified in 5.5.11 and 5.5.12.

Table C.11 – ArgBlock GWM_PDControlPrio

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D07
2	Count	This element provides the number N of following PDControlPrios	Unsigned8	0x01 to 0xFF
3	PDControlPrioSetList	List of PDControlPrioSet as defined in Table C.12	Array[N] of PDControlPrioSet	n/a

The length of this ArgBlock is 3 octets plus the length of the PDControlPrioSetList.

Table C.12 –PDControlPrioSet

Octet Offset	Element name	Definition	Data type	Values
0	TargetClientID	ClientID associated to the priority. 0: all clients (only for SGI_SetPDControlPrio) 1 to 255: specific client	Unsigned8	0x00 to 0xFF
1	TargetModuleNumber	ModuleNumber associated to the priority. 0: all module numbers (only for SGI_SetPDControlPrio) 1 to 255: specific module number	Unsigned8	0x00 to 0xFF
2	TargetPortNumber	PortNumber associated to the priority. 0: all port numbers (only for SGI_SetPDControlPrio) 1 to 255: specific port number	Unsigned8	0x00 to 0xFF
3	PDControlPrio	Priority 0: highest priority 255: lowest priority	Unsigned8	0x00 to 0xFF

The length of a PDControlPrioSet is 4 octets.

C.9 GWM_Address

The ArgBlock GWM_Address defined in Table C.13 is used as REQ ArgBlock by the services SGI_GetAccessRights and SGI_GetPDCControlPrio specified in 5.5.7 and 5.5.12.

Table C.13 – ArgBlock GWM_Address

Octet Offset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D08
2	TargetClientID	0: all clients 1 to n: specific client	Unsigned8	0x00 to 0xFF
3	TargetModuleNumber	0: all modules 1 to n: specific module	Unsigned8	0x00 to 0xFF
4	TargetPortNumber	0: all ports 1 to n: specific port	Unsigned8	0x00 to 0xFF

The length of this ArgBlock is 5 octets.

C.10 GWM_ReleasePDCControl

The ArgBlock GWM_releasePDCControl defined in Table C.14 is used by the service SGI_ReleasePDCControl specified in 5.5.13.

Table C.14 – ArgBlock GWM_ReleasePDCControl

OctetOffset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D09
2	TargetModuleNumber	0: all modules 1 to n: specific module	Unsigned8	0x00 to 0xFF
3	TargetPortNumber	0: all ports 1 to n: specific port	Unsigned8	0x00 to 0xFF

The length of this ArgBlock is 4 octets.

C.11 GWM_Status

The ArgBlock GWM_Status defined in Table C.15 provides the current status of the exclusive access paths.

Table C.15 – ArgBlock GWM_Status

OctetOffset	Element name	Definition	Data type	Values
0	ArgBlockID	Unique ID	Unsigned16	0x7D0A
2	OwnPDCControlPrio	Priority 0: highest priority 255: lowest priority	Unsigned8	0x00 to 0xFF
3	PDOUTCurrentClient	0: no exclusive access 1 to 255: ClientID	Unsigned8	0x00 to 0xFF
4	PDOutCurrentPrio	Priority	Unsigned8	0x00 to 0xFF
5	AcycCurrentClient	0: no exclusive access 1 to 255: ClientID	Unsigned8	0x00 to 0xFF
6	AcycExclusiveCause	Cause 0: No exclusive access 1: Block transfer 2: BLOB transfer 3: User request 4 to 255: reserved	Unsigned8	0x00 to 0xFF

516 The length of this ArgBlock is 7 octets.

517

Annex D (normative)

SGL ArgBlock ErrorTypes

D.1 SGL ArgBlock ErrorTypes

D.1.1 SGL associated ErrorTypes

The GWM returns ErrorTypes within a negative response (Result (-)) while performing an SGL service or propagated SMI service. Table D.1 shows the SGL specific ErrorTypes.

Table D.1 – SGL ArgBlock related ErrorTypes

Incident	Error Code	Additional code	Name
ClientID invalid	0x41	0x40	CLIENT_ID_INVALID
ModuleNumber invalid	0x41	0x41	MODULE_NUMBER_INVALID
Access not granted	0x41	0x42	ACCESS_NOT_GRANTED
Access temporarily not granted	0x41	0x43	ACCESS_TEMPORARILY_NOT_GRANTED
GWM-Address not valid	0x41	0x44	GWM_ADDRESS_INVALID
Access rights too strict	0x41	0x45	ACCESS_RIGHTS_TOO_STRICT

An SGL or propagated SMI Service may also return ErrorTypes as specified in Table C.3 in [1]. The priority of the ErrorTypes responded by SGL or SMI services is defined in Table D.2 with descending priority.

Table D.2 – Priority of ArgBlock ErrorTypes

Priority	Name	Error Code	Additional code	Definition
1 (highest)	CLIENT_ID_INVALID	0x41	0x40	See D.1.2
2	MODULE_NUMBER_INVALID	0x41	0x41	See D.1.3
3	PORT_NUM_INVALID	0x40	0x11	See D.1.4
4	ARGBLOCK_NOT_SUPPORTED	0x40	0x01	See D.1.5
5	ACCESS_NOT_GRANTED	0x41	0x42	See D.1.6
6	ACCESS_TEMPORARILY_NOT_GRANTED	0x41	0x43	See D.1.7
7	ARGBLOCK_LENGTH_INVALID	0x40	0x34	See D.1.8
8	ARGBLOCK_INCONSISTENT	0x40	0x02	See D.1.9
9	MEMORY_OVERRUN	0x40	0x06	See D.1.10
10	GWM_ADDRESS_INVALID	0x41	0x44	See D.1.11
11 (lowest)	ACCESS_RIGHTS_TOO_STRICT	0x41	0x45	See D.1.12

The SGL ErrorTypes are specified with further restrictions or implementation hints in the following clauses.

D.1.2 CLIENT_ID_INVALID

This ErrorType shall be used if the provided ClientID is unknown to the SGL.

D.1.3 MODULE_NUMBER_INVALID

This ErrorType shall be used if the provided ModuleNumber is unknown to the SGL.

D.1.4 PORT_NUM_INVALID

This ErrorType shall be used if the provided PortNumber is unknown to the SGI or SMI, see C.4.8 in [1].

D.1.5 ARGBLOCK_NOT_SUPPORTED

See C.4.2 in [1].

D.1.6 ACCESS_NOT_GRANTED

This ErrorType shall be used if an SGI or SMI service does not have the right to perform the request

D.1.7 ACCESS_TEMPORARILY_NOT_GRANTED

This ErrorType shall be used if an SMI service is requested while the single resource is occupied by another client, see 5.4.4 and 5.4.5.

D.1.8 ARGBLOCK_LENGTH_INVALID

See C.4.10 in [1].

D.1.9 ARGBLOCK_INCONSISTENT

See C.4.3 in [1].

D.1.10 MEMORY_OVERRUN

See C.4.7 in [1] and this ErrorType shall be used if SGI_SetAccessRights or SGI_GetAccessRights require more memory for receiving or responding access rights lists.

D.1.11 GWM_ADDRESS_INVALID

This ErrorType shall be used if an address triple defined in a GWM_Address defined in C.9 is unknown to the SGI.

D.1.12 ACCESS_RIGHTS_TOO_STRICT

This ErrorType shall be used if SGI_SetAccessRights does not leave at least one ClientID for further configuration of rights.

Annex E
(normative)

Test strategies

E.1 General

to be defined at end of specification to define coverage of tests for gateway implementations
###

E.2 Test ??

to be defined at end of specification to define coverage of tests for gateway implementations
###

Bibliography

- [1] IO-Link Community, *IO-Link Interface and System*, V1.1.5, October 2025, Order No. 10.002
- [2] PI specification, IO-Link Integration – *Edition 2, Version 1.1*, December 2019, Order No. 2.832
- [3] ODVA, *CIP Common Specification volume 7C*, November 2019
- [4] EtherCat Technology Group, *Modular Device Profile – Part 6220: IO-Link Master*, V1.0.5, April 2017, Order No. ETG 5001.6220
- [5] OPC Foundation Companion Specification, *OPC UA for IO-Link*, Version 1.0, December 2018
- [6] IO-Link Community, *JSON – Integration for IO-Link, Version 2.0*, March 2026, Order No. 10.222
- [7] IO-Link Community, *IO-Link Profile BLOB Transfer & Firmware Update Specification, Version 1.1.5*, October 2025, Order No. 10.082
- [8] IO-Link Community, *IO-Link Common Profile, Version 1.2.1*, October 2025, Order No. 10.072
-

© Copyright by:

IO-Link Community

c/o PROFIBUS Nutzerorganisation e.V.

Ohiostrasse 8

76149 Karlsruhe

Germany

Phone: +49 (0) 721 / 98 61 97 0

Fax: +49 (0) 721 / 98 61 97 11

e-mail: info@io-link.com

<http://www.io-link.com/>



IO-Link