

The logo for TEConcept, featuring the text "TEConcept" in a sans-serif font. The "TE" is in black, and "Concept" is in a lighter blue-grey. A red horizontal line is positioned beneath the text, starting under the "T" and ending under the "t".

TEConcept

IODD Studio User's Manual



IODD Studio User's Manual



Document ID	MA_TECU_051_01
Author	Peter Zay;Pal Kergyik
Issue Date	2022-04-22
Status	Draft
Approved by	
Recipient / Access	For Publication
Version	V 1.0.0



1

2 Revision History

Version	Date	Description	Edited by	Status
0.1.0	2022-03-02	Initial	PA	Approved
0.2.0	2022-03-17	Review	PZ	Review
0.3.0	2022-03-23	Updated based on review by PZ.	PA	Rework
1.0.0	2022-04-22	Release candidate version	PZ	Rework
				Select...
				Select...
				Select...
				Select...
				Select...
				Select...
				Select...

3

4

5



6	Table of Contents	
7	1 Scope	5
8	1.1 Purpose of this Document	5
9	1.2 Applicable Documents	5
10	1.3 Terms, Acronyms and Abbreviations	5
11	2 Overview	6
12	3 Quickstart	8
13	4 Manual	12
14	4.1 Main Menu bar	12
15	4.2 Management bar	13
16	4.3 Main area Tabs	14
17	4.3.1 Start	14
18	4.3.2 IODD	16
19	4.3.3 Settings	35
20	4.3.4 Analyzer Settings	37
21	4.3.5 IODD Finder	38
22	4.3.6 IOLFW Packager	39
23	4.3.7 Creating templates	40
24	4.4 Status bar	41
25	4.4.1 Analyzer messages	41
26	4.4.2 IODD Checker messages	41
27	5 Issue reports	43
28		
29		



1 Scope

1.1 Purpose of this Document

This document includes all technical descriptions of the software design of the . The design is based on the specifications and/or requirements listed in the requirements section of the corresponding project management document.

1.2 Applicable Documents

Ref-ID	Doc-ID	Document Title	Date	Version
1	IO-Device-Desc-Spec_10012_V113_Jan21.pdf	Specification related to IO-Link Interface and System Specification V1.1.3 and IODD Schemas V1.1	January 2021	1.1.3
2	IOL-Profile_Firmware-Update_V11_10082_Sep19.pdf	IO-Link Profile BLOB Transfer & Firmware Update Specification	September 2019	1.1
3				
4				

References to applicable documents point to the Ref-ID in this format: [1].

1.3 Terms, Acronyms and Abbreviations

Term/Acronym/Abbreviation	Description
IODD	IO-Link Device Description
PDCT	Port and Device Configuration Tool
ISDU	Indexed Service Data Unit



2 Overview

An IODD (IO Device Description) is a set of files that formally describes a device e.g., IO-Link Device. The IODD is created by the device vendor. The IODD can be used by engineering tools for PLCs and/or PDCTs for the purpose of identification, configuration, definition of data structures for Process Data exchange, parameterization, and diagnosis decoding of a particular Device. The set of files consists of the main IODD file, optional language files and optional picture files. An IODD is mandatory for each IO-Link Device.

The IODD file is an XML file, creating and editing manually those files are difficult. The IODD Studio Software's intention is to help developing these files, it gives a user-friendly interface where the user only needs to specify the values and the Application will automatically create the IODD file.

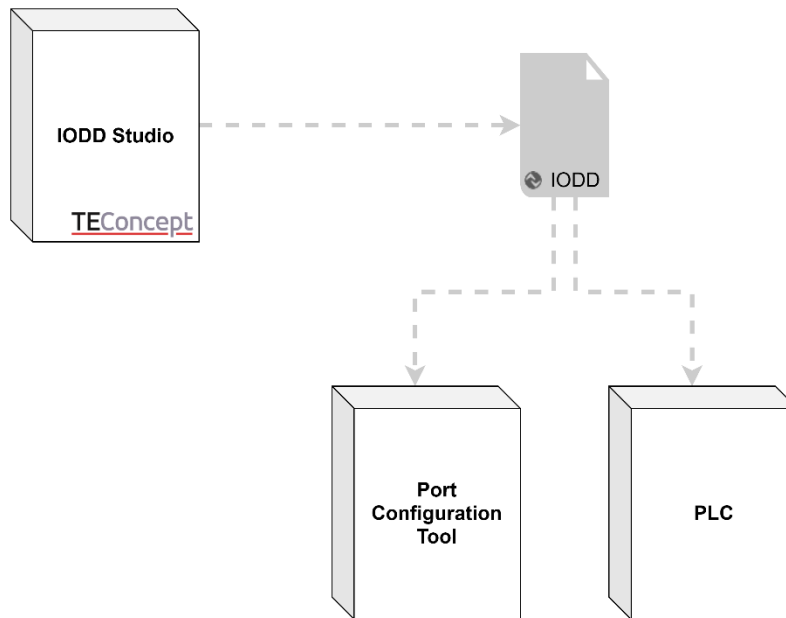


Figure 1 Overview

The IODD Studio Software's gives complete power on the IODD creation it allows to create / build an IODD in any point and not just if the created IODD is valid. The tool only validates upon editing the most basic issues, like text input when numeric value is expected, however multiple analyzers are running in the background that gives feedback about the found errors, warnings or notifications using this information the implementer of the IODD file can modify on the values.

It is important to note that even if the built-in analyzers are reporting some issues the user has the rights to ignore it and create the IODD. The built-in analyzers are in some cases stricter than the IODD specification, eventually the external IODD Checker validates the created IODD so it must consider it as valid and not the Application.

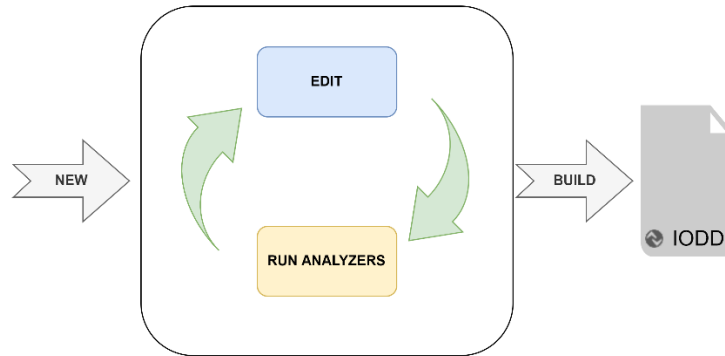


Figure 2 IODD Editing Workflow

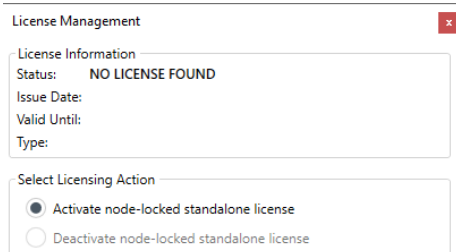
The desired workflow of the IODD Studio Application is that a new IODD is created (or an existing IODD is imported) it is being edited in the Application, the analyzers are triggered upon property change events and any time the user is able to create the IODD.



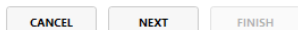
3 Quickstart

The following steps are intended as a brief guide to get the IODD Studio up and running:

1. First install the IODD Studio using the provided .msi package. Double clicking this executable shall start the installer.
2. After starting the freshly installed IODD Studio the first step is to activate your license.
3. Select “Help” \ “License Key Manager” from the menu bar.



[View EULA...](#)



4. The above window shall appear. ***If for some reason the “Status” row shows that there is an activated license, then there is nothing further to do.*** Otherwise,
5. The “Activate node-locked standalone license” shall be automatically selected. Click the “Next” button.
6. If a message box appears with the message “Please restart as administrator to access license management!” then please start the application as administrator and repeat the steps starting from Step 3.



79 7. The following dialog box shall appear next:

- 80
81 8. If you do not yet have your license, then please follow the displayed instructions: email the “Work-
82 station ID” to the info@teconcept.hu e-mail address. Otherwise skip this step and proceed to step 9.
83 Please note that:

- 84 a. Licenses are issued per computer and the same license key can not be used to activate two dif-
85 ferent computers.
86 b. Significant alterations to your computer’s hardware/software configuration may require you to
87 obtain another license key from TEConcept.



9. If you already have your license key, then please insert (paste) it into the License Key text box and press the “Next” button.

License Management

Workstation ID

IMQyx
+OUXrvMDWSLW5sG3QIVa0nHMFQwCJypsLQRZn46TDkI5THOUfqC
IZd4U2g59jD4ndzAtqvSYXrWowF0yw==

COPY

Please email the above Workstation ID to TEConcept to receive a valid license.
Please include your name, company and contact information to the following email address:
info@teconcept.hu
They will provide a License Key that will enable the Software.

License Key

Paste the provided License Key into the field:

I2lnbmF0dXJlVmFsdWU+PC9TaWduYXR1cmU+PC9MaWNlbnNlPg==

CANCEL NEXT FINISH

10. If you receive a message box with the text “License Invalid!” then make sure that you have pasted the correct license exactly as it appears into the License Key text box. If the issue persists, please contact TEConcept at info@teconcept.hu regarding your licensing issue.
11. If a correct license is inserted the following dialog shall appear with the text “LICENSE IS VALID”:

License Management

License Operation Status

LICENSE IS VALID

CANCEL NEXT FINISH

12. Click “FINISH”. Your license is now active.



13. You can check its status by clicking “Help” / “License Key Manager” on the menu bar. A license dialog with an activated license looks like the following

[View EULA...](#)



4 Manual

This section describes the user interface elements and their basic functionality in the IODD Studio Application. The following image demonstrates the basic user interface of the IODD Studio Application.

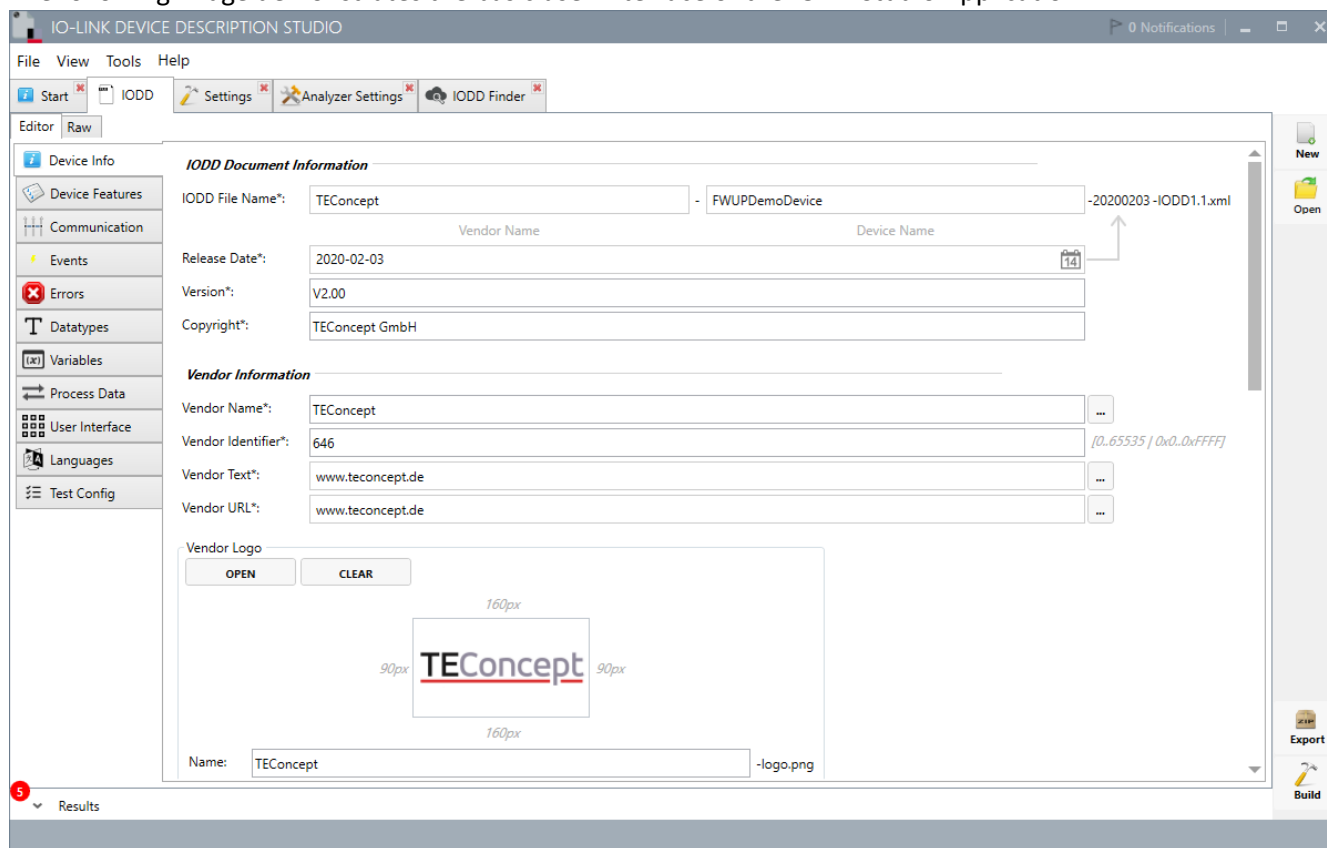


Figure 3 IODD Studio User Interface

4.1 Main Menu bar

On top of the Application's main window the Menu Bar is located. The Menu Bar consist of the following elements: File, View, Tools, and Help.

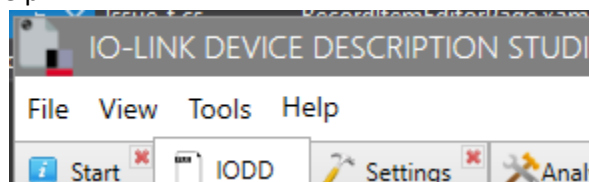


Figure 4 Main Menu bar

The following sub-menus are available for the main menu bar:

- File
 - New -> Creates new project, brings up the new project dialog. See [4.2.1.1.](#)
 - Open -> Open existing project.



- Save -> Saves the current state in a binary form with .isproj extension, this format is only usable by the IODD Studio Application.
- Save as -> Allows saving the currently opened project under a different name.
- Exit -> Closes the application.
- View
 - Checker configuration -> Opens and jumps to the “Analyzer Settings” page. For more information see [4.2.4](#)
 - Start page -> Opens and jumps to the “Start” page. For more information see [4.2.1](#)
 - IODD Finder -> Opens and jumps to the “IODD Finder” page. For more information see [4.2.5](#)
 - IOLFW Packager -> Opens and jumps to the “IOLFW Packager” page. For more information see [4.2.6](#)
 - Settings -> Opens and jumps to the “Settings” page. For more information see [4.2.3](#)
- Tools
 - Create Template -> Opens the “New Template” window. For more information see [4.2.7](#)
 - Import from IODD -> Opens the “Import Window” window. For more information see [4.2.8](#)
- Help
 - License Key Manager -> Opens the “License Management” window. See the [Quickstart](#) section for the license activation process.
 - About TEConcept IODD Studio -> Opens the about window.

4.2 Management bar

The IODD management bar is located in the right side of the main window.

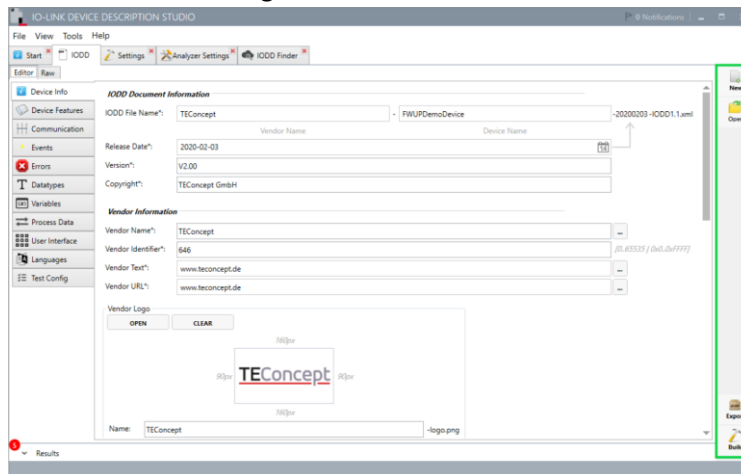


Figure 5 Management bar

The following functions are available in this panel:

- New – provides the same functionality as the File / New menu item.
- Open – provides the same functionality as the File / Open menu item.
- Export – exports the IODD file with the additional language files and images to a selected directory.



- Build – builds the IODD into the hard drive to the location set in the Settings / Output Files / Directory Name.

4.3 Main area Tabs

This chapter describes the main area's tab items, the following list specifies the available tab items:

- Start – this tab item is the start page, the user can create new project, open an existing project, import an IODD or select a recently used project in tis tab item.
- IODD – this user interface element represents the IODD's properties, this view is used to edit the IODD file.
- Settings – contains Application related settings.
- Analyzer Settings – the user is able to configure the background analyzers in this tab item.
- IODD Finder – the user can download IODDs from the IODD Finder data base in this tab item.
- IOLFW Packager – this tab item is used to create IO-Link Device Firmware Download package. See [2].

4.3.1 Start

This view is shown to the user when the Application starts, the user can select the desired action from this start view.

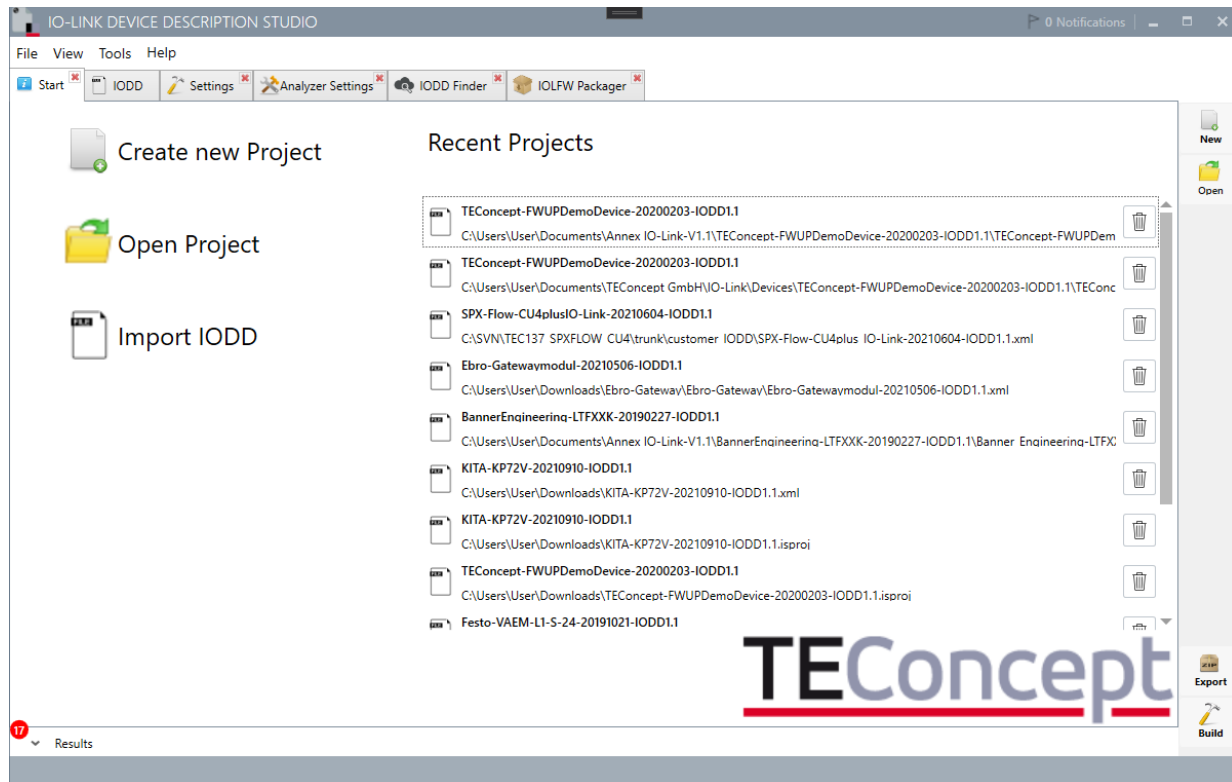


Figure 6. Start Page

The start page on the left-hand side shows the three operations that can be used to start the editing procedure:

- By creating a new project.
- By opening an existing project.



- By importing an existing IODD.

The right-hand side of the start page shows recently opened projects or IODDs if there are any the user can click on them to load the recently used project / IODD.

4.3.1.1 Create New Project Dialog

The New Project dialog is used for selecting the project type (standard, wireless device or wireless bridge) and it allows the specification of some of the most common device profiles (BLOB Transfer, Firmware Download profiles). Depending on the selected project type and selected feature the Application will set up the most basic requirements for the selected project type.

From the project types only one can be selected, and any number of features can be included. In case a project template is selected the features are ignored.

Figure 7. New Project Dialog

For creating templates see section: [Creating templates](#).



4.3.2 IODD

This page appears only if a new project has been started or an existing IODD / project has been loaded. This view is used for editing the IODD using both a visual and a text editor.

4.3.2.1 Views

The IODD page is shown when a new project is created or an existing IODD / project is opened. It contains both the “Editor” and the “Raw” sub-page. The Editor sub-page maps the IODD elements to a visual editor. The mapping tries to be one to one whenever that is possible or desirable.

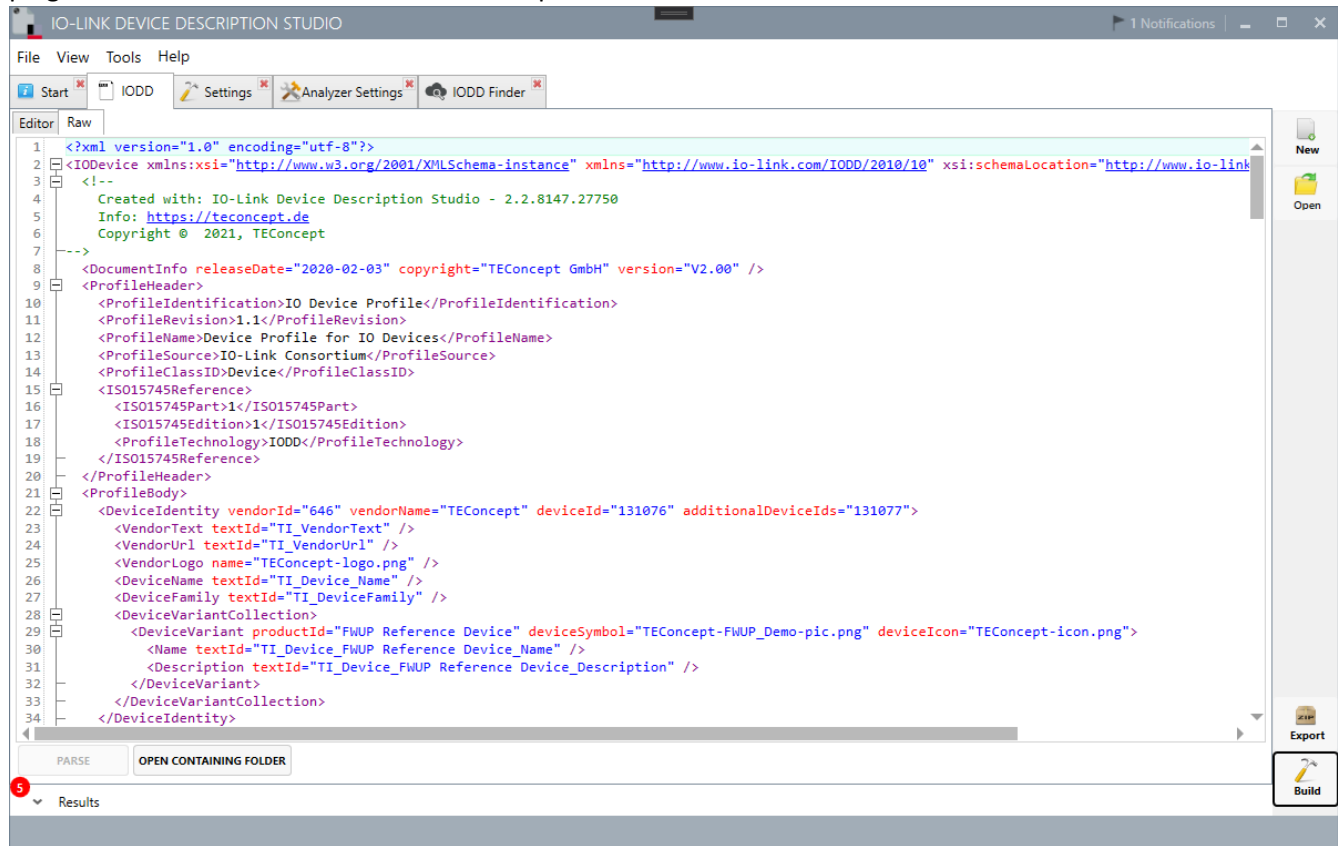


Figure 8 IODD page / Raw sub-page



The IODD page contains two views “Editor” and “Raw” on two sub-pages. The “Editor” page contains the intelligent editor, and the “Raw” page contains the raw contents of the IODD which can be viewed and edited. This IODD can always be generated/updated by pressing the “Build” button and parsed if modified using the “PARSE” button. This allows for the editing of the IODD both in the XML format and by using the intelligent editor. The tabs located on the left-hand side of this page correspond to the features of the IODD format and they are used to edit/view various parts of the IODD. The intelligent editor has the following tabs:

- Device Info: Visual editing of the DeviceIdenty element of the IODD. See [\[1\]/7.4.](#)
- Device Features: Common feature and profile support can be set here. See [\[1\]/7.5.1.](#)
- Communication: Visual editing of the PhysicalLayer element of the IODD. See [\[1\]/7.6.](#)
- Events: Standard and vendor specific event specification. See [\[1\]/7.5.7.](#)
- Errors: Standard and vendor specific error specification. See [\[1\]/7.5.6.](#)
- Datatypes: Visual editing of the DatatypeCollection element of the IODD. See [\[1\]/7.5.2.](#)
- Variables: Standard and vendor specific variable definition. See [\[1\]/7.5.4.](#)
- Process Data: The process data collection can be visually specified on this tab. See [\[1\]/7.5.5.](#)
- User Interface: User interface, roles editing. See [\[1\]/7.5.8.](#)
- Languages: Specify languages other than English. See [\[1\]/7.7.](#)
- Test Config: Visual editing of the test configuration. See [\[1\]/Figure 56.](#)

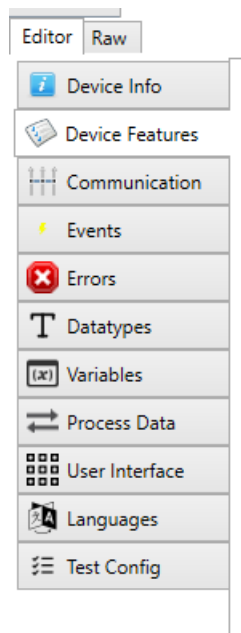


Figure 9. IODD page / Editor sub-page / List of tabs



4.3.2.2 Device Info

Required and optional information relating to the IODD, Vendor, Device and Device variants can be viewed and edited here. The three areas of this page map directly to the XML elements specified in the IODD specification.

IODD Document Information

IODD File Name*: - -20220311 -IODD1.1.xml

Release Date*: 2022-03-11

Version*: V1.0

Copyright*:

Vendor Information

Vendor Name*: ...

Vendor Identifier*: 0x0000 [0..65535 | 0x0..0xFFFF]

Vendor Text*: ...

Vendor URL*: ...

Figure 10. IODD/Editor/Device Info/IODD Document Information and Vendor Information area

The data displayed in the “IODD Document Information” area maps directly to the DocumentInfo element in the IODD. See [\[1\]/7.3.1](#).

The vendor name used in the file name(s) (IODD, images, languages) can be different than the Vendor Name inserted into the IODD, the IODD Studio will use the Vendor Name provided in the IODD File Name input field as the default vendor name part of the additional files (IODD, images, languages).

Device Identification

Device Name*: FWUP Demo Device ...

Device Identifier*: 131076 [1..16777215 | 0x00001..0]

Additional Device Ids

DEVICE ID
131077

NEW EDIT REMOVE

Device Family*: Demo Device ...

Device Variants*

Count:

PRODUCT ID	NAME	ICON	SYMBOL	DESCRIPTION
FWUP Reference Device	Demo FW 1			Reference Device demonstrating Firmware Update with TEConcept Device Stack

Figure 11 . IODD/Editor/Device Info/IODD Device Information area

In the “Device Identification” Device related information could be specified, for more about the elements see [\[1\]/7.4](#).



216

217

Note: the IODD Studio provides the same four action when working with collections:

218

- New – for adding a new element into the collection.

219

- Edit – for editing an already existing element in the collection.

220

- Remove – deleting the selected element in the collection.

221

- Clone – for cloning an element already present in the collection, this is useful when differences between two elements are minor thus this function speeds up the creation of the new element.

222

223

224

Note: by double clicking in the cells where the “Lock” symbol is not present in the column header gives direct editing possibility for the values without the need of clicking on the “Edit” button.

225



4.3.2.3 Device Features

Supported device features and profiles can be selected on this editor component. Features include block parametrization, data storage and access locks support. Profiles include, among others, the BLOB Transfer, Firmware Download Profiles. The IODD Studio analyzer engine is aware of the changes imposed by the profiles. The Features element of the IODD is generated based on the settings specified here. How features and profiles are specified is described in [\[1\]/7.5.1](#). For more information about the firmware update and BLOB transfer features see [\[2\]](#).

Supported Device Features

☐ Block Parametrization
☐ Data Storage Supported
☐ Access locks support

☐ Data Storage Access Lock
☐ Parameter Access Lock
☐ Local Parametrization Access Lock
☐ Local user Interface Lock

Supported Profiles

☐ BLOB Transfer Profile (PID: 0x0030)
☐ Firmware Download Profile (PID: 0x0031)
☐ Identification & Diagnosis Profile (PID: 0x4000)
☐ IO-Link Safety Profile (PID: 0x4001)
☐ Smart Sensor Profile (PID: 0x0001)

☐ Analog (PIDs: 0x8000, 0x8002)
☐ Binary (PIDs: 0x8000, 0x8001)
☐ Analog + Binary (PIDs: 0x8000, 0x8001, 0x8002)

PID	NAME

NEW

EDIT

REMOVE

Figure 12. IODD/Editor/Device Features



4.3.2.4 Communication

The IODD Studio allows the used Communication Network Profile definition in this editor component. The physical layer properties can be configured according to the selected network profile type (wired or wireless). For more see the specification [\[1\]/7.6](#).

Network Profile Type*: Wired

Physical Layer

Minimum Cycle Time*: 0 μ s

0 - AFAP (As Fast As Possible)
400 μ s - 6300 μ s | In steps of 100 μ s
6400 μ s - 31600 μ s | In steps of 400 μ s
32000 μ s - 132800 μ s | In steps of 1600 μ s

Bitrate*: COM1 (4,8 kbit/s)

TBIT time: 208,33 μ s

☐ Standard Input / Output Supported (SIO)*

Connector(s)

Count: 0

PRODUCT ID(S)	CONNECTION TYPE	CONNECTION SYMBOL	DESCRIPTION

NEW EDIT CLONE REMOVE

Figure 13. IODD/Editor/Communication/Physical Layer Area

For a wired Device in the Physical Layer area the user can specify the cycle time, bitrate (COM speed) whether standard input / output is supported and the used connector type.



Protocol Layer

☒ Indexed Service Data Unit (ISDU) Supported

☐ Compatible with V1.0
Whether the device is compatible with IO-Link revision 1.0, i.e. also runs on a V1.0 IO-Link Master.
If checked, two IODD is created: V1.1 and V1.0.1.

Preoperate Mode

M-sequence Type:

On-request Data Length:

Operate Mode

M-sequence Type:

On-request Data Length:

Process Data Input Length:

Process Data Output Length:

Coded M-sequence Capability

M-sequence Capability:

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	0	0	0	0	0	0	1
RESERVED		PREOPERATE M-SEQUENCE TYPE		OPERATE M-SEQUENCE TYPE		ISDU	

Figure 14. IODD/Editor/Communication/Protocol Layer Area

The “Protocol Layer” area contains the settings for ISDU support, V1.0 compatibility and M-Sequence type selection in Operate and Preoperate modes. The “mSequenceCapability” attribute is automatically calculated based on these settings. The tool automatically shows the binary encoding of the m-sequence capability and its numerical value.



4.3.2.5 Events

On this page the user can configure the IO-Link Events supported by the IO-Link Device, it is possible to include standard events and it is also possible to define vendor specific events. Device related IO-Link Events are defined in the specification [1] (see 7.5.7).

The following image shows the Standard Events:

☐ INCLUDE	CODE	TYPE	NAME	DESCRIPTION
☐	0	Notification	No malfunction	
☐	4096	Error	General malfunction	Unknown error
☒	16384	Error	Temperature fault	Overload
☐	16912	Warning	Device temperature overrun	Clear source of heat
☐	16928	Warning	Device temperature underrun	Insulate device
☐	20480	Error	Device hardware fault	Exchange device
☐	20496	Error	Component malfunction	Repair or exchange
☒	20497	Error	Non-volatile memory loss	Check batteries
☐	20498	Warning	Batteries low	Exchange batteries
☐	20736	Error	General power supply fault	Check availability
☐	20737	Error	Fuse blown or open	Check fuse switch or exchange fuse
☐	20752	Warning	Primary supply voltage overrun	Check valid voltage range
☒	20753	Warning	Primary supply voltage underrun	Check valid voltage range
☐	20754	Warning	Secondary supply voltage fault (Port Class B)	Check valid voltage range
☐	24576	Error	Device software fault	Check firmware revision
☐	25376	Error	Parameter error	Check datasheet and values

Figure 15 IODD/Editor/Events/Standard Events

In this tab item the user is able to include or exclude Standard IO-Link Events, these events are specified in the IODD-StandardDefinitions1.1.xml file (location of this file is configured in the [Settings](#) tab item) the above listed standard IO-Link Events are parsed from the XML file the user can not extend this list only including / excluding is available for each event.

It is also possible for the user to define vendor specific events on the tab named "Vendor specific Events" as seen below:



267

Standard Events
Vendor specific Events

Vendor Specific Events

Count: 2
IMPORT

CODE	TYPE	NAME	DESCRIPTION
36350	Error	Test Event A	
36351	Error	Test Event B	

NEW
EDIT
CLONE
REMOVE

Figure 16 IODD/Editor/Events/Vendor Specific Events

268

269

270

271 This page can be used to specify vendor specific events. The following functions are available in this view:

- 272 • New – for creating a new Event.
- 273 • Edit – editing an existing Event definition.
- 274 • Clone – cloning an existing event, the user can spare time with this functionality for example if the new
- 275 event is similar to the existing one.
- 276 • Remove – Deletes the selected event.

277 **Note:** it is possible for the User to import event definition(s) from a different IODD by using the “Import” but-

278 ton and selecting the IODD where the events are already defined



4.3.2.6 Errors

On this page the user can configure the IO-Link errors supported by the IO-Link Device, it is possible to include standard errors and it is also possible to define vendor specific errors. Device related IO-Link errors are defined in the specification [1] (see 7.5.6).

The following image shows the Standard Errors:

Standard Errors **Vendor specific Errors**

Standard Error Types

Count: 17

☐ INCLUDE	CODE	ADDITIONAL CODE	NAME	DESCRIPTION
☐	128	0	Device application error - no details	Service was denied by the technology-specific application. No
☒	128	17	Index not available	Read or write access attempt to a non-existing index.
☒	128	18	Subindex not available	Read or write access attempt to a non-existing subindex of ar
☒	128	32	Service temporarily not available	Parameter not accessible due to the current state of the techr
☒	128	33	Service temporarily unavailable - local control	Parameter not accessible. The device is currently in an ongoin
☒	128	34	Service temporarily unavailable - device control	Parameter not accessible. The technology-specific application
☒	128	35	Access denied	Write access to a read-only parameter or read access to write
☒	128	48	Parameter value out of range	Written parameter value is outside of the permitted value ran
☒	128	49	Parameter value above limit	Written parameter value is above its specified value range
☒	128	50	Parameter value below limit	Written parameter value is below its specified value range
☒	128	51	Parameter length overrun	Written parameter is longer than specified.
☒	128	52	Parameter length underrun	Written parameter is shorter than specified.
☒	128	53	Function unavailable	Written command is not supported by the technology-specifi
☒	128	54	Function temporarily unavailable	Written command is unavailable due to the current state of th
☒	128	64	Invalid parameter set	Written single parameter value collides with other existing pa
☒	128	65	Inconsistent parameter set	Parameter set inconsistencies at the end of block parameter t

INCLUDE **EXCLUDE**

Figure 17 IODD/Editor/Errors/Standard Errors

In this tab item the user is able to include or exclude Standard IO-Link Errors, these errors are specified in the IODD-StandardDefinitions1.1.xml file (location of this file is configured in the [Settings](#) tab item) the above listed standard IO-Link errors are parsed from the XML file the user can not extend this list only including / excluding is available for each error.

It is also possible for the user to define vendor specific errors on the tab named “Vendor specific Errors” as seen below:



294

Standard Errors
Vendor specific Errors

Vendor Specific Error Types

Count: 1
IMPORT

Code	ADDITIONAL CODE	NAME	DESCRIPTION
129	255	Test Error	

NEW
EDIT
CLONE
REMOVE

295

296

Figure 18 IODD/Editor/Errors/Vendor Specific Errors

297

This page can be used to specify vendor specific errors. The following functions are available in this view:

298

- New – for creating a new Error.

299

- Edit – editing an existing Error definition.

300

- Clone – cloning an existing error, the user can spare time with this functionality for example if the new error is similar to the existing one.

301

302

- Remove – Deletes the selected error.

303

Note: it is possible for the User to import error type definition(s) from a different IODD by using the “Import”

304

button and selecting the IODD where the error types are already defined



4.3.2.7 Datatypes

This page allows the creation, editing, cloning, and removing of user defined data types for used inside the IODD. The created data types can be later referenced by various variable or process data definitions. Information shown here corresponds to the DatatypeCollection element inside the IODD, see [\[1\]/7.5.2](#). The following functions are available in this view:

- New – for creating a new data type.
- Edit – editing an existing data type definition.
- Clone – cloning an existing data type, the user can spare time with this functionality for example if the new error is similar to the existing one.
- Remove – deletes the selected data type.

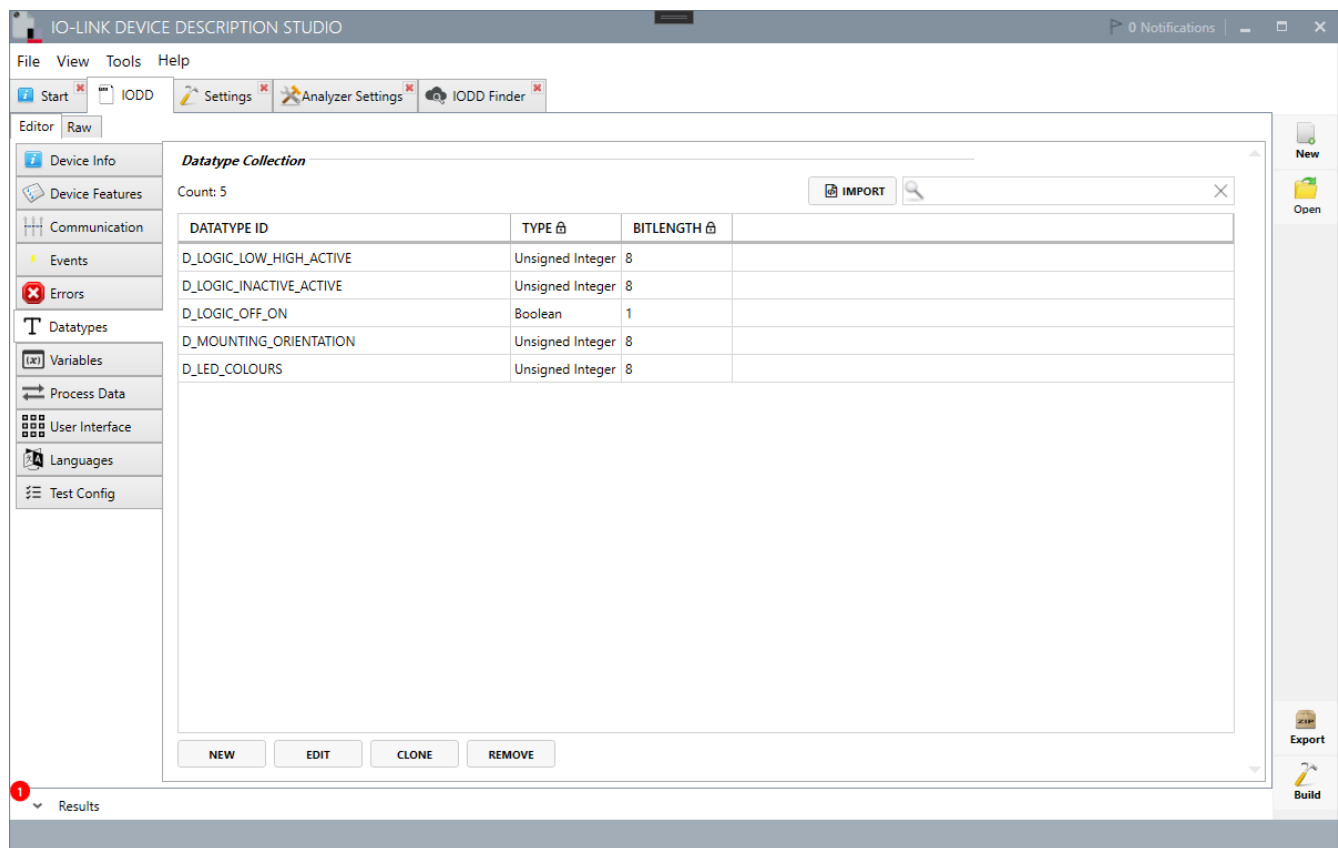


Figure 19 IODD/Editor/Datatypes

Note: it is possible for the User to import data type definition(s) from a different IODD by using the “Import” button and selecting the IODD where the data types are already defined.



4.3.2.8 Variables

The variables tab of the editor can be used to insert both Standard Variables as defined in [1]/7.5.4 and vendor specific variables into the IODD. The standard variables are specified in the IODD-StandardDefinitions1.1.xml file (location of this file is configured in the [Settings](#) tab item).

The IODD Studio offers a dedicated editor view for the Standard Variables.

Standard Variables

- ☒ V_DirectParameters_1
- ☒ V_DirectParameters_2
- ☒ V_SystemCommand

Standard System Commands

INCLUDED	Value	Name
☒	129	Application Reset
☒	130	Restore Factory Settings
☒	131	Back-to-box

System Commands

NAME	VALUE
Leave failsafe state	160
Reset customer parameters	161

Standard Variable Details:

Id: *V_SystemCommand* Dynamic: ☐

Name: *System Command* Modifies Other Variables: ☒

Index: *0x0002* Access Right: *WO*

Datatype: *Unsigned Integer*

Buttons: NEW, EDIT, REMOVE

Figure 20 IODD/Editor/Variables/Standard Variables

The following figure shows the vendor specific variable definitions tab:



Standard Variables
Vendor specific Variables

Vendor Specific Variable Collection

Count: 61

IMPORT

INDEX	ID	NAME	RIGHTS	DYN.	MOD	EXCL	DATATYPE	
25	V_FunctionTag	Function Tag	Readable / Writable	☐	☐	☐	String	
26	V_LocationTag	Location Tag	Readable / Writable	☐	☐	☐	String	
65	V_SAMPLE_MODE	Sampling rate of measurement (only logging mode)	Readable / Writable	☐	☒	☐	Unsigned Integer	Select
66	V_SAMPLE_RATE	Sampling rate in manual mode (logging mode only)	Readable / Writable	☒	☐	☐	Unsigned Integer	Select
77	V_log_digit_output_end_pos_closed	Logic digital output end position "CLOSED"	Readable / Writable	☒	☐	☐	D_LOGIC_LOW_H	Invert
78	V_switching_point_sens_pos_closed	Switching point sensor position "CLOSED"	Readable / Writable	☒	☐	☐	Unsigned Integer	Switch
79	V_log_digit_output_end_pos_open	Logic digital output end position "OPEN"	Readable / Writable	☒	☐	☐	D_LOGIC_LOW_H	Invert
80	V_switching_point_sens_pos_open	Switching point sensor position "OPEN"	Readable / Writable	☒	☐	☐	Unsigned Integer	Switch
89	V_acceleration_sens_value	Acceleration sensor value	Read Only	☒	☐	☐	Unsigned Integer	Acceler
90	V_max_acceleration_value	Max. acceleration value	Read Only	☒	☐	☐	Unsigned Integer	Maxi
91	V_calibr_acceleration_sens	Calibration acceleration sensor	Readable / Writable	☒	☒	☒	D_LOGIC_OFF_ON	Acceler
92	V_act_deact_led_disp	Activate / Deactivate LED display	Readable / Writable	☒	☐	☐	D_LOGIC_INACTIV	Inter
93	V_led_colour_pos_closed	LED colour position CLOSED	Readable / Writable	☒	☐	☐	D_LED_COLOURS	LED

NEW
EDIT
CLONE
REMOVE

Figure 21 IODD/Editor/Variables/Vendor specific Variables

The following functions are available in this view:

- New – for creating a variable definition.
- Edit – editing an existing variable definition.
- Clone – cloning an existing variable, the user can spare time with this functionality for example if the new variable is similar to the existing one.
- Remove – Deletes the selected variable.

Note: by right clicking on the selected variable element the User has the possibility to move the data type defined for the variable to the data type collection.

Note: it is possible for the User to import variable definition(s) from a different IODD by using the "Import" button and selecting the IODD where the variables are already defined.



4.3.2.9 Process Data

On this page the process data collections can be defined. The process data elements are containing the process data input and process output element definition alongside with the condition element, for more about the process data see [\[1\]/7.5.5](#).

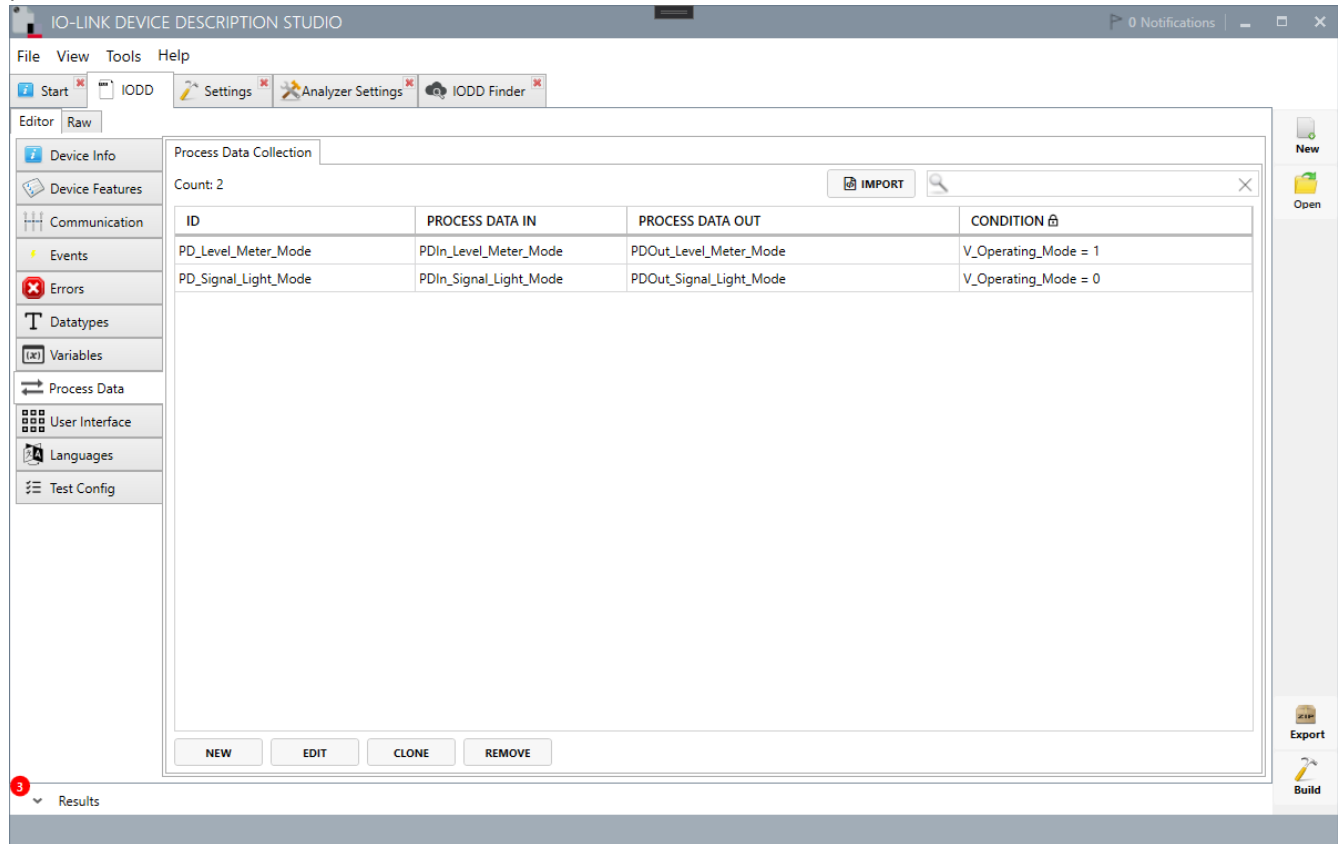


Figure 22 IODD/Editor/Process Data

The following functions are available in this view:

- New – for creating a new process data collection.
- Edit – for editing an existing process data collection.
- Clone – for cloning an existing process data collection: time can be saved by cloning an existing, similar process data collection instead of making one from scratch.
- Remove – Deletes the selected process data collection.

Note: it is possible for the User to import process data definition(s) from a different IODD by using the “Import” button and selecting the IODD where the process data elements are already defined.



4.3.2.10 User Interface

User interface, menus for the various roles (Observer, Maintenance, and Specialist) can be created, edited on this editor component. See [\[1\]/7.5.8](#) for more information about the User Interface element of the IODD.

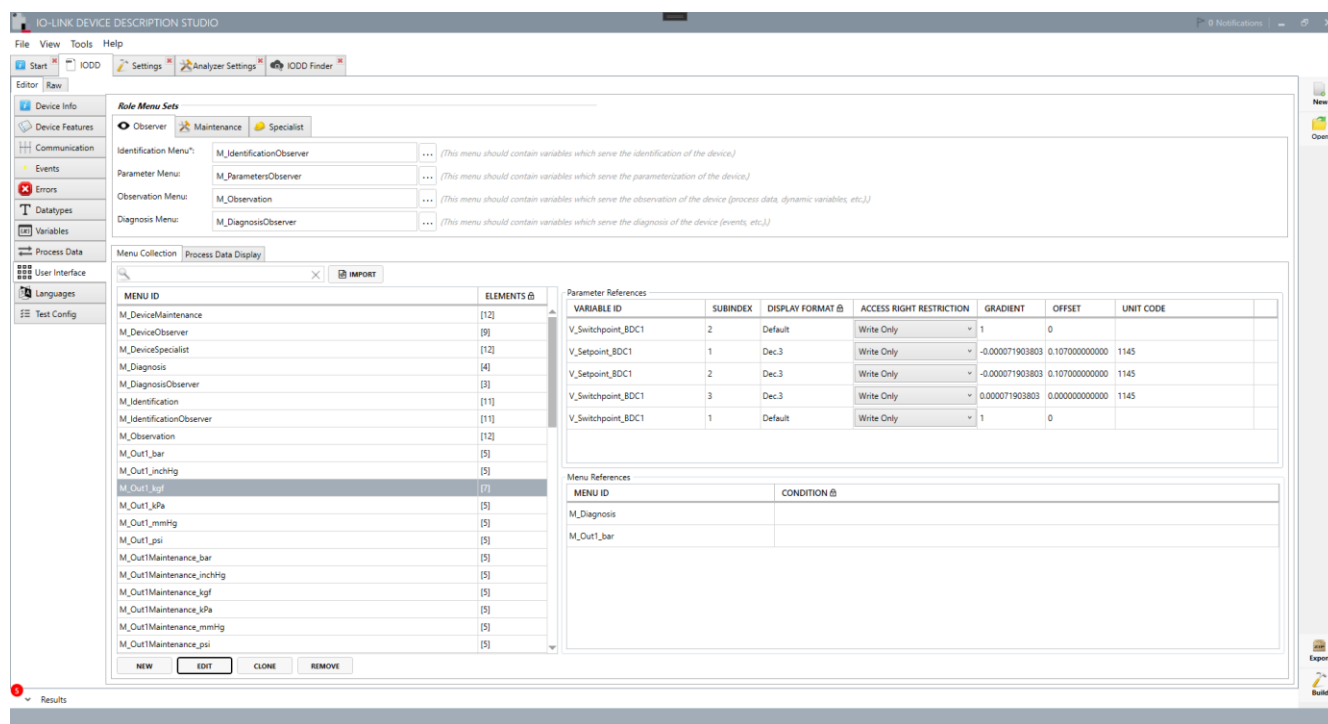


Figure 23 IODD/Editor/User Interface

In this editor component three different User Interface element can be configured:

- Role Menu Sets – the User can configure the top level menus referenced in various roles (Observer, Maintenance and Specialist) by typing in the menu identifier or by pressing on the selection button (with three dots).
- Menu Collection – the User can create, remove or edit menu elements.
- Process Data Display – special display format can be specified for each process data input or output element.

Note: by right clicking on a menu element the User can see the full menu reference depth in the current menu in a graph view (“Reveal referenced menus”) or the menus referencing the current element (“Show Dependency”).

Note: it is possible for the User to import menu(s) from a different IODD by using the “Import” button and selecting the IODD where the menu elements are already defined.



4.3.2.11 Languages

The IODD uses key – value pairs to represent every text value in the IODD file, the defined texts are then referenced by their unique identifier, this solution gives the possibility to translate the IODD to different languages. This tab is used to specify alternative languages other than the mandatory English according to [11/5.1](#), [11/5.2](#).

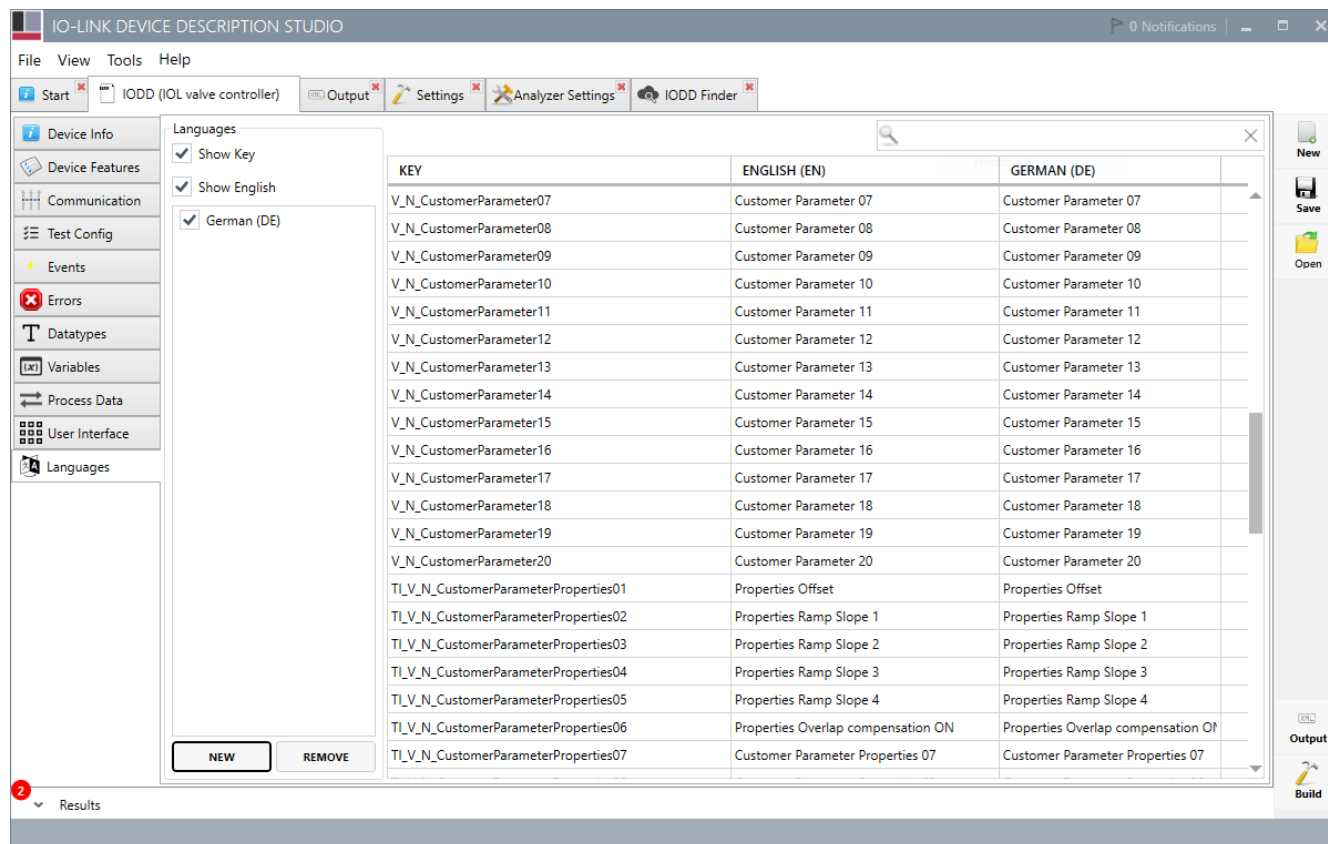


Figure 24 IODD/Editor/Languages

In this view the user can see the texts and their key used across the IODD file, it is possible to add a new language definition as well, and the user even has the possibility to use an existing language as the base of the new language definition.

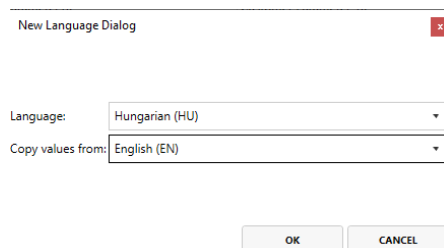


Figure 25 New Language Dialog

The IODD Studio allows the possibility to reuse an existing text as well in the input fields where text reference is used under the hood.



Device Name*: IO-Link Generic Device



Device Identifier*: 000000

Figure 26 Referencing an existing Text

By pressing the button with three dots the IODD Studio will show a dialog where every text value will be shown and the User can select an already existing text and the key will be automatically reused therefore the text value will not be duplicated in the external text collection element of the IODD.

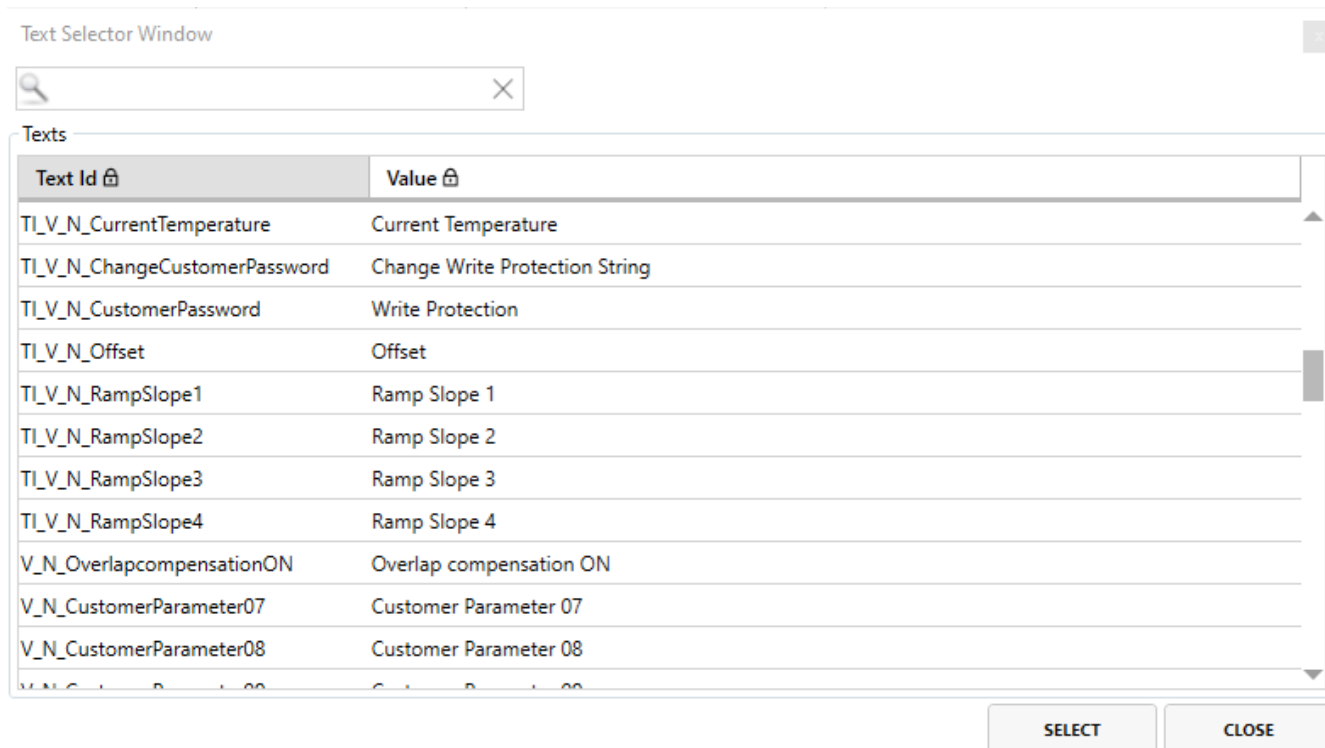


Figure 27 Text Selector Dialog



4.3.2.12 Test Config

The test configuration element stores the information necessary for the protocol tester Applications. Test configuration data corresponding to the XML elements described by [1]/Figure 56 and chapter [1]/7.6 are specified here.

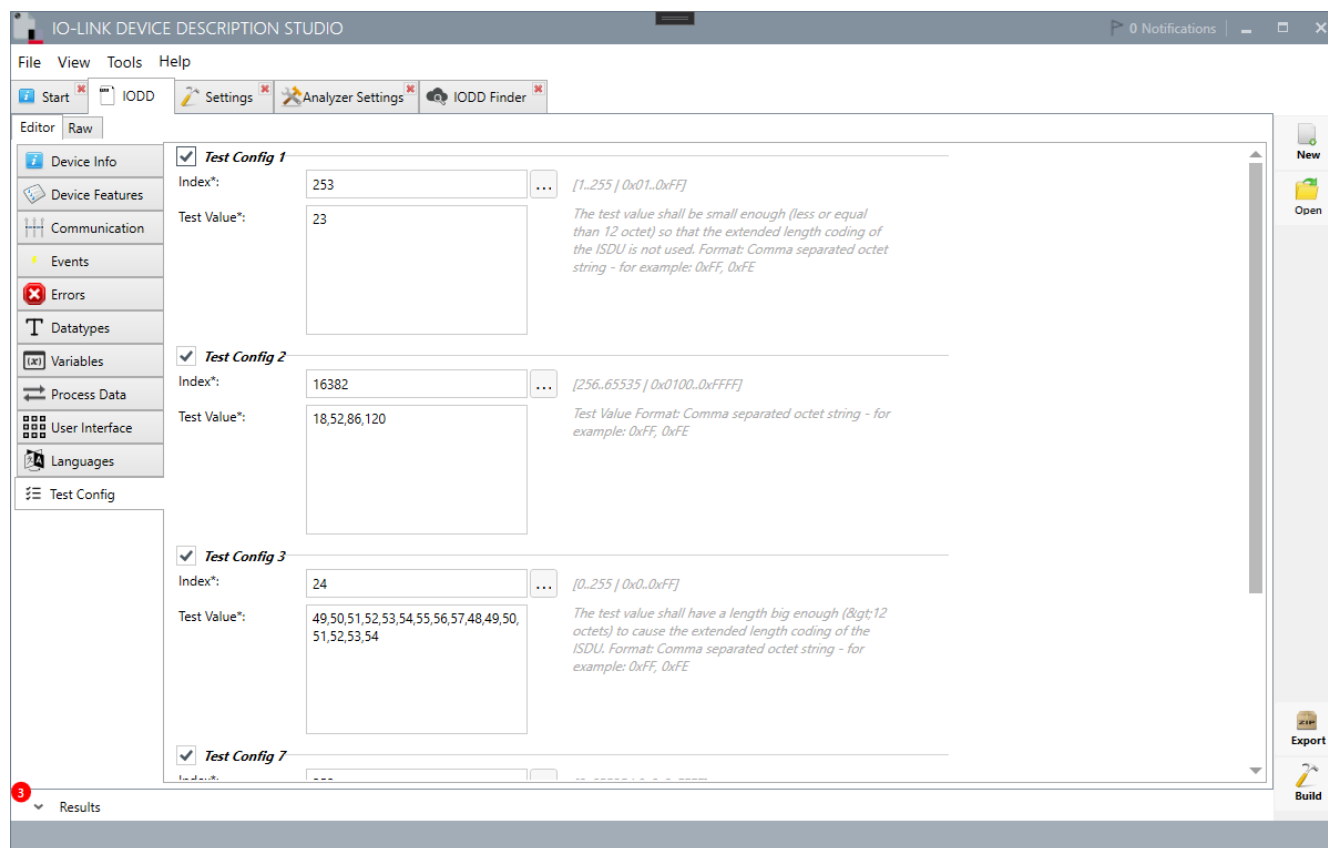


Figure 28 IODD/Editor/Test Config

The User can enable / disable test configuration elements and also specify the values used by the tester Applications in this editor component.



4.3.3 Settings

Application related settings can be found in this view.

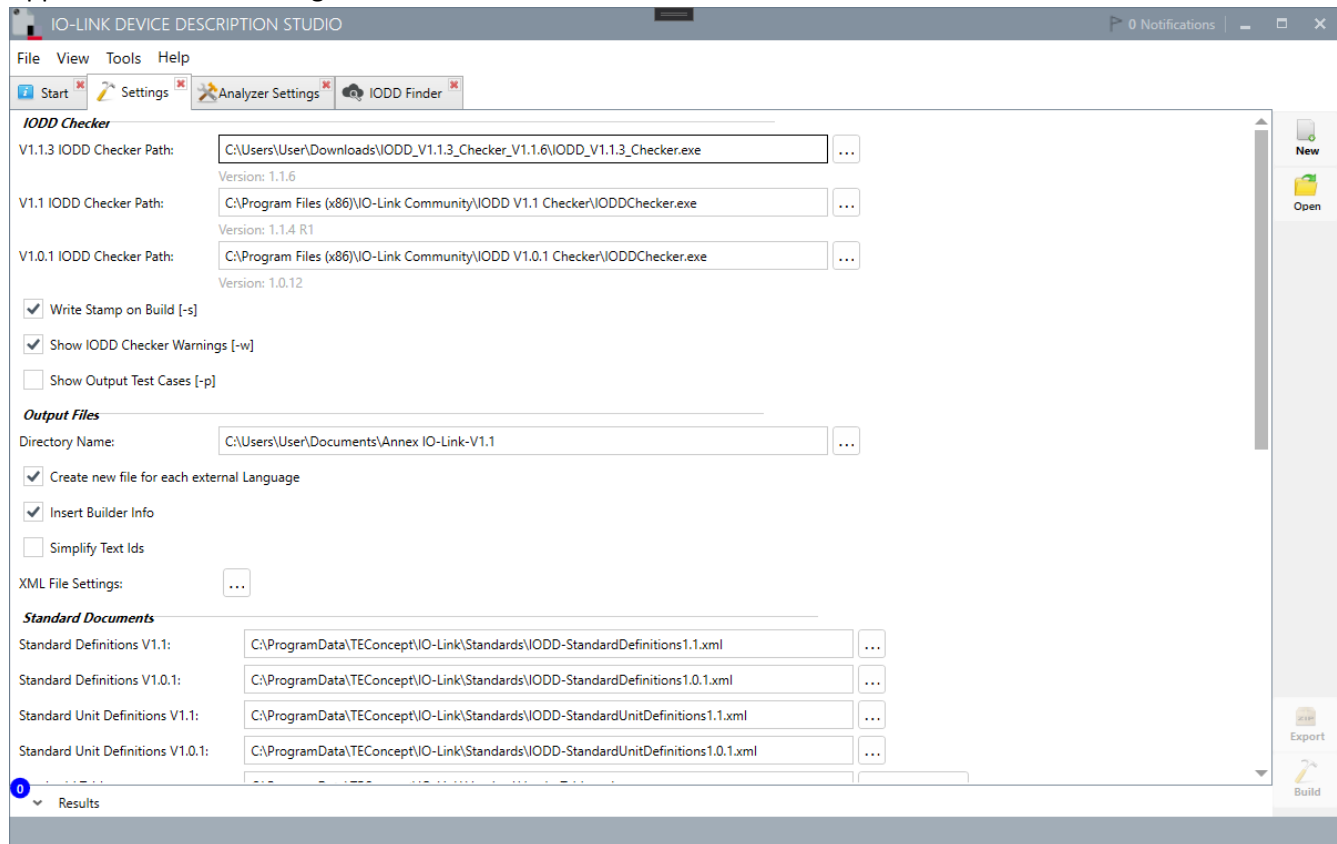


Figure 29 Settings view

On the settings page the following preferences can be set:

- The location of the IODD checker executables for various IODD versions.
- IODD Checker configuration
 - Whether warnings from the checker shall be shown or not.
 - Whether output test cases should be shown by the IODD Checker.
 - Writing of the stamp on build.
- Output files
 - The output directory of the generated IODDs
 - Whether to create new XML files for the additional languages
 - Whether to insert builder information into the created IODD
 - Whether to simplify the used text identifiers in the IODD
 - Additional XML file related configurations
- The location of the IODD Standard Documents.
- Download directory for the IODD Finder.
- Preferred number format for displaying numbers (hexadecimal / decimal).
- Prefixes for the various elements of the IODD.



- 430 ○ Datatype Id Prefix
- 431 ○ Variable Id Prefix
- 432 ○ Process Data Id Prefix
- 433 ○ Menu Id Prefix
- 434 ● Theme of the UI.
- 435



4.3.4 Analyzer Settings

The IODD Studio runs the analyzer in the background to detect errors and warn about potential issues while editing. The settings are available in raw JSON format and via a specified user interface.

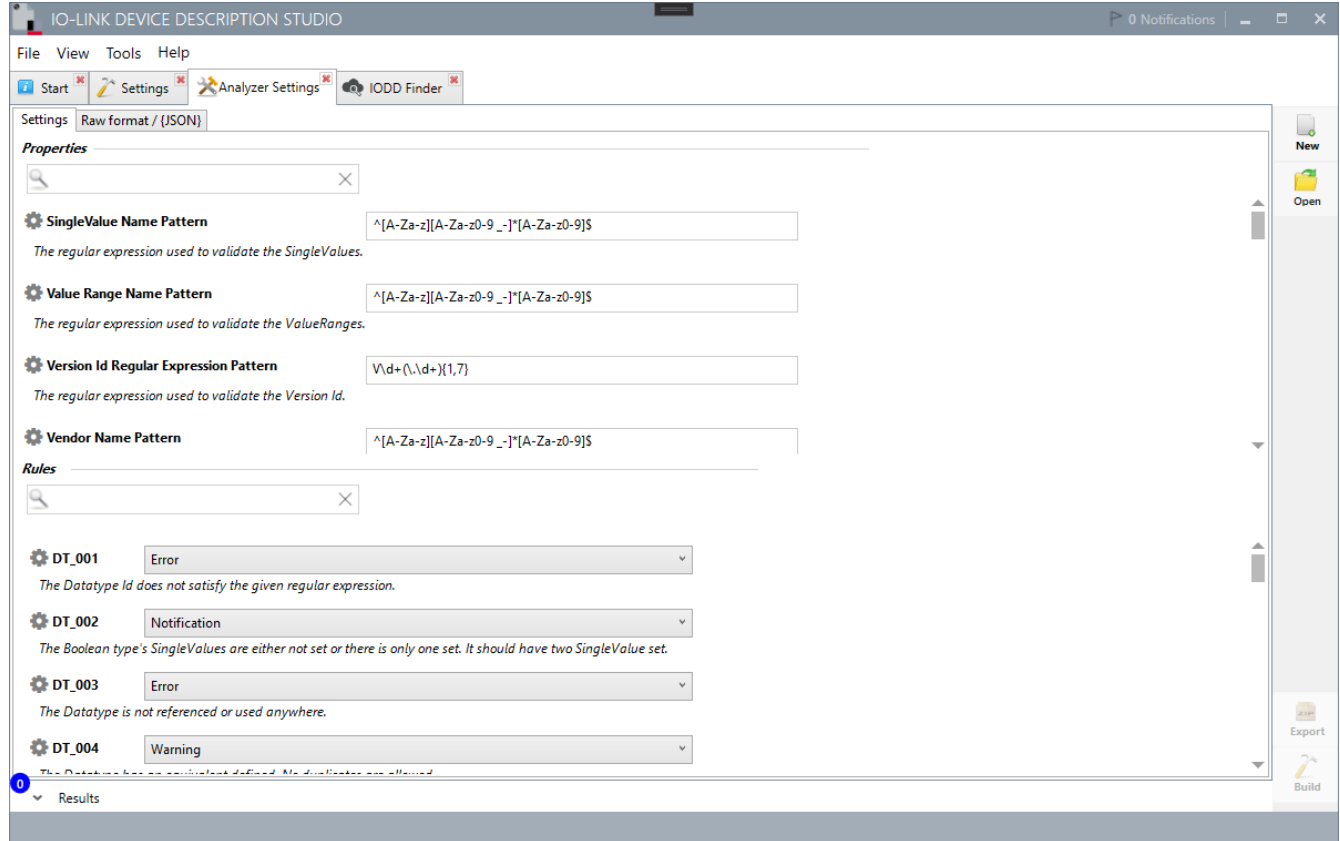


Figure 30 Analyzer settings

This page can be used to configure the analyzer. This includes the configuration of:

- Various patterns for detecting if a string is correct according to the IODD specification.
- Allowed event ranges.
- Maximum allowed instances for various IODD elements.
- Allowed number ranges for various IODD/device attributes.
- The analyzer engine rules which device the severity of certain violations.

As it was previously mentioned the user can create stricter rules than required by the IODD Specification and also the user can enforce company rules (variable naming rule etc.) for the created IODDs by the Application.



4.3.5 IODD Finder

The IODD Finder can be used to find IODDs belonging to a given vendor in the IODD database. The user can specify different filters to execute more specific search for the desired IODD, like device identifier, product name, revision etc.

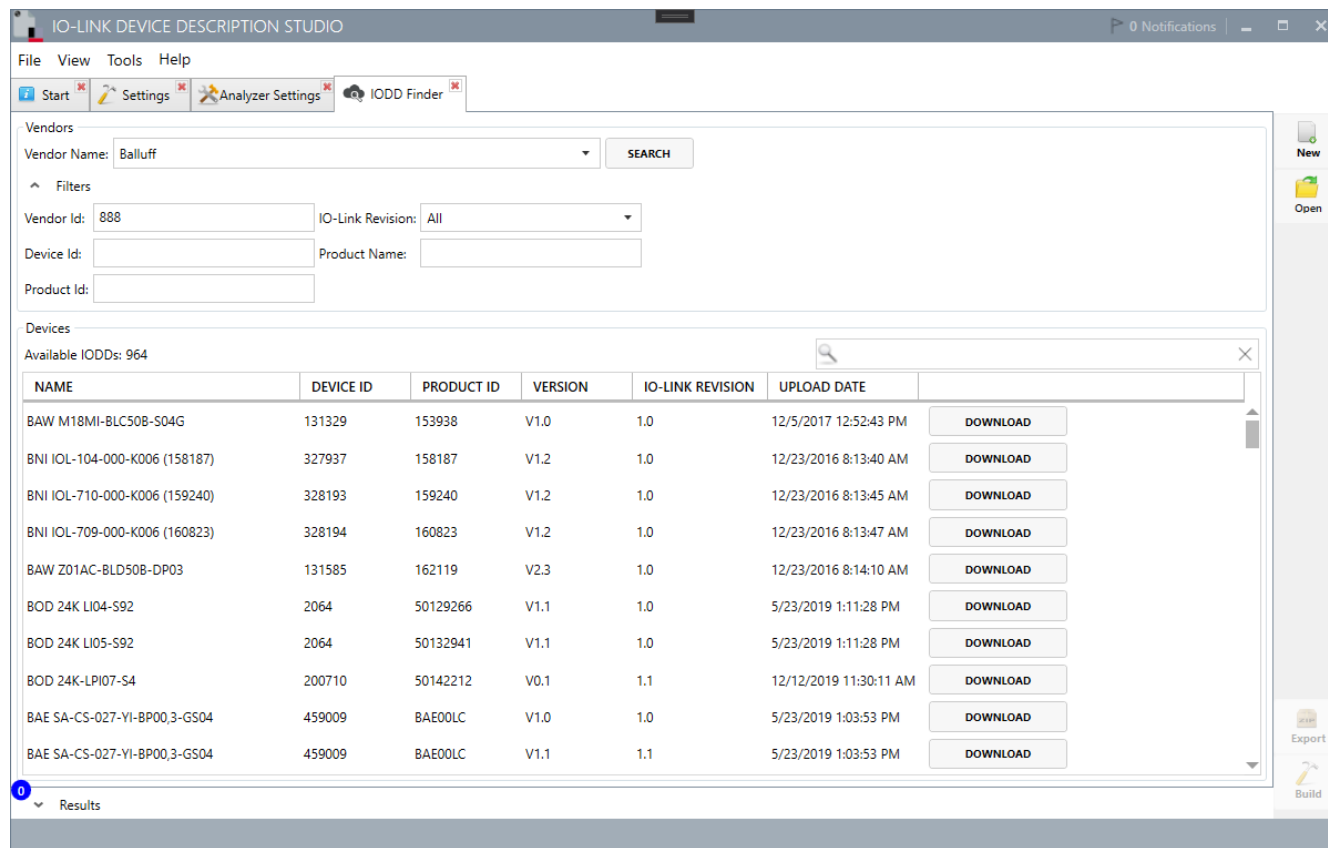


Figure 31 IODD Finder view

The selected IODD can be downloaded and imported into the IODD Studio. The downloaded IODDs are placed to the directory configured in the Settings view.



4.3.6 IOLFW Packager

This page can be used to generate IOLFW packages for distributing firmware for IO-Link Devices supporting the Firmware Update profile. See [2].

Figure 32. IOLFW Packager Page

The user can create a new package or open an existing IOLFW package as well, in case of open operation the IODD Studio will extract the content of the compressed package into the directory configured in the Settings view.



4.3.7 Creating templates

The IODD Studio is able to serialize the actual properties and create a template from the current state (see [Create New Project Dialog](#)). This is useful for example when the User filled out all the Vendor specific information like logo, name etc. and want to use it as the base of the future IODDs.

New templates can be created by selecting the “Create Template” command from the “Tools” menu on the menu bar. The “Create Template” dialog can only be opened after an IODD / project is opened or new project is started as it can be used to save the current IODD as a template.

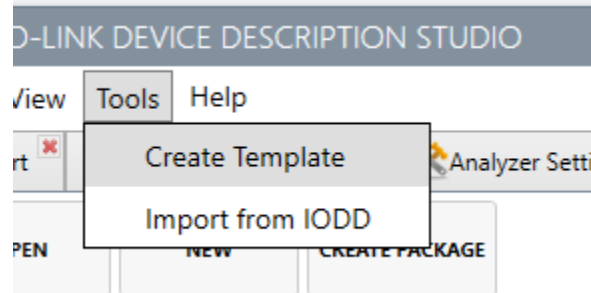


Figure 33. The Create Template Menu Item

After opening the dialog, you can specify a name and a short description for the template as it can be seen on the image below:

New Template

Name*: Generic TEConcept IODD Template

Description: A generic IODD Studio template for TEConcept Devices

SAVE CANCEL

Figure 34. New Template Dialog

Templates are stored in the directory configured in the [Settings](#) view.



4.4 Status bar

The status bar contains the collapsible Results table containing notifications, warnings, and errors from the analyzer engine and the IODD checker on two separate tabs.

4.4.1 Analyzer messages

When the user edits the parameters of the IODD it internally triggers the built-in analyzers, and the output of these analyzers are shown in this view. As it was already mentioned the built-in analyzers are in some cases stricter than the IODD Specification or the user sets up stricter validation rules (see [Analyzer Settings](#)) so it is not a requirement for creating the IODD to not have errors, at the end the IODD Checker must consider the IODD as valid.

Code	Type	Module	Message
E_007	Error	Events/Test Event A (EventCode: 36350)	The description for the event is not specified, the description of the event is mandatory when backward compatibility is set.
E_007	Error	Events/Test Event B (EventCode: 36351)	The description for the event is not specified, the description of the event is mandatory when backward compatibility is set.
COM_028	Error	Communication	The specified cycle time (500 µs) is shorter than the shortest possible cycle time (5800 µs) calculated for the actual M-sequence.
ME_048	Error	User Interface	Standard Variable 'V_DeviceAccessLocks' with read-write access right and not excluded from data storage is not used in sequence.
FWDL_009	Error	Variables/0x43BE (17342)(V_HW_ID_Key)	The variable's data type has invalid fixed length set. It should be: 16 octet.
COM_007	Warning	Communication/FWUP Reference Device	The connection's description element is not specified.
ME_021	Warning	User Interface/M_Identification	The menu (M_Identification) is referenced in the Observer role and has parameter(s) with write access consider limiting the rights.
ME_021	Warning	User Interface/M_Parameter	The menu (M_Parameter) is referenced in the Observer role and has parameter(s) with write access consider limiting the rights.
DI_015	Notification	IODD Document Information	The document info's release date (2020-02-03) is not today (2022-04-22).

Figure 35 Analyzer results

Description of the displayed columns are as follows:

- Code: Code of the rule that is being triggered/violated.
- Type: Error, Warning, Notification. The severity of the issue.
- Module: which part of the editor / analyzer module the error is originating from. The location of the error.
- Message: description of the issue.

For configuring the analyzer engine, please see the [Analyzer Settings](#) section.

4.4.2 IODD Checker messages

On the IODD Checker tab the messages from the external IODD checker are displayed. Every time the “Build” button is pressed the IODD is created in the output directory and the IODD Checker is called with the newly created IODD.

The IODD Studio Application will automatically call the right IODD Checker version according to the actual IODD (V1.0.1, V1.1 or V1.1.3).

The IODD Studio Application will also pass the additional files to the IODD Checker (language files, V1.0.1 IODD – if compatibility is set) and the validation messages of these files will be concatenated to the original IODD’s validation message.

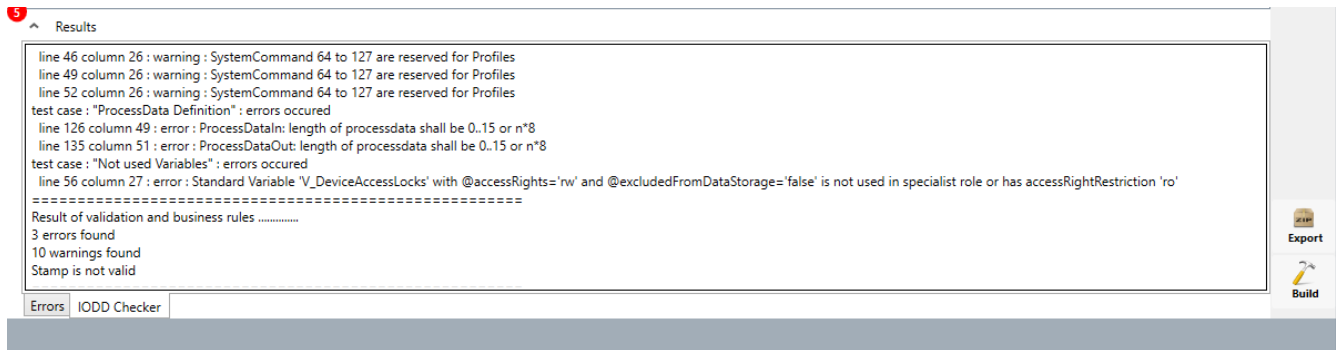


Figure 36 IODD Checker output

IODD Checker related configuration is available in the Settings tab page (see [Settings](#)).



5 Issue reports

The about box shows the complete version information of the Application.



Figure 37 About box

It is important to include this information in case of issuing a change request or issue report.