



FIELDCOMM GROUP™

*Connecting the World of
Process Automation*



FDI Usability Style Guide

FCG AG12080 (FDI-2080)

Edition 1.0.1b

10 Sep 2018

RELEASED

Document Distribution / Maintenance Control / Document Approval

To obtain information concerning document distribution control, maintenance control, and document approval please contact FieldComm Group at the address shown below.

Copyright © 2018 FieldComm Group

This document contains copyrighted material and may not be reproduced in any fashion without the written permission of FieldComm Group.

Trademark Information

FieldComm Group™, FOUNDATION™ Fieldbus and HART-IP™ are trademarks, and HART®, *WirelessHART*®, ROM® and SIF® are registered trademarks of FieldComm Group, Austin, Texas, USA. Any use of these terms hereafter in this document, or in any document referenced by this document, implies the trademark/registered trademark. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information, contact FieldComm Group at the address below.



FieldComm Group
Attention: President and CEO
9430 Research Boulevard
Suite 1-120
Austin, TX 78759, USA
Voice: (512) 792-2300
FAX: (512) 792-2310

<http://www.fieldcommgroup.org>

Intellectual Property Rights

The FieldComm Group (the Group) does not knowingly use or incorporate any information or data into the HART, FOUNDATION Fieldbus and FDI protocol standards, which the Group does not own or have lawful rights to use. Should the Group receive any notification regarding the existence of any conflicting private IPR, the Group will review the disclosure and either (A) determine there is no conflict; (B) resolve the conflict with the IPR owner; or (C) modify the standard to remove the conflicting requirement. In no case does the Group encourage implementers to infringe on any individual's or organization's IPR.

TABLE OF CONTENTS

1	Introduction.....	7
1.1	Abbreviations	7
1.2	Terms & definitions.....	8
1.2.1	FDI specific terms.....	8
1.2.2	Data type names and references to data types	8
1.2.3	Vocabulary for requirements	8
1.2.4	Specific formatting.....	8
1.3	Normative references	9
1.4	Purpose	9
1.5	Field of application	10
1.5.1	UID designers	10
1.5.2	UIP designers.....	10
1.6	How to use this document	10
2	Usability goals	11
2.1	Specified users and their goals	11
2.2	Context of use	11
2.3	Effectiveness, efficiency and satisfaction	11
3	General usability heuristics.....	12
3.1	Visibility of system status	12
3.2	Match between system and the real world	12
3.3	User control and freedom.....	12
3.4	Consistency and standards	12
3.5	Error prevention	12
3.6	Recognition rather than recall	12
3.7	Flexibility and efficiency of use.....	13
3.8	Aesthetic and minimalist design.....	13
3.9	Help users to recognize, diagnose, and recover from errors	13
3.10	Help and documentation	13
4	User views	14
4.1	Maintenance	14
4.2	Specialist	14
5	Use cases	15
5.1	Device commissioning	15
5.2	Function test	15
5.3	Device calibration.....	15
5.4	Condition monitoring	15
5.5	Data synchronization	15
5.6	Device replacement	15
5.7	Device decommissioning.....	15
6	Platform capabilities and constraints	16
6.1	Hosting.....	16
6.1.1	Navigation element	16

6.1.2	Action element.....	17
6.1.3	Automated generation of action elements →UID.....	17
6.1.4	Services for StandardUIActions →UIP.....	17
6.2	Input and output devices.....	17
6.2.1	Keyboard	17
6.2.2	Pointing devices (mouse or touch screen).....	17
6.2.3	Screen	17
6.2.4	Printer.....	17
7	Top-level menus	18
7.1	Mandatory top-level menu entries →UID	18
7.1.1	Operate.....	18
7.1.2	Diagnostics.....	18
7.1.3	Device settings.....	18
7.2	Structures below the top-level menus →UID.....	18
7.2.1	Usability aspects of the menu structure.....	18
7.2.2	Representation	19
8	Dialog design.....	20
8.1	Fully labelled variables →UIP	20
8.2	Variables with predefined values	20
8.2.1	Mutually exclusive choices →UID	20
8.2.2	Mutually exclusive choices →UIP.....	21
8.2.3	Multiple non-exclusive choices →UID	21
8.2.4	Multiple non-exclusive choices →UIP	22
8.3	Tooltips	22
8.3.1	Tooltips →UID	22
8.3.2	Tooltips→UIP	23
8.4	Design for resizability →UIP	23
8.5	Text fonts →UIP.....	23
8.6	Contrast of text and images →UIP	23
8.7	Colors →UIP.....	23
8.8	Tables and graphs →UID	24
8.9	Modal and non-modal dialogs	24
8.10	Wizards.....	24
8.10.1	Wizard state machine →UID.....	24
8.10.2	Wizard implementation →UIP	25
9	Parameter handling.....	26
9.1	Workflows and caches	26
9.1.1	Activate UI	26
9.1.2	Edit variables.....	26
9.1.3	Update device cache.....	26
9.1.4	Transferring data from/to the device.....	26
9.2	Variable value states	27
9.2.1	Evaluation of variable value state →UID.....	27
9.2.2	Evaluation of variable value state →UIP.....	27

10	Appendix	29
10.1	Dictionary	29
10.1.1	Menu labels	29
10.1.2	Action area labels.....	29
10.1.3	Variable value states	31
11	Bibliography	32

Revision Log

Version	Originator	Date	Change Note/History/Reason
0.9.0	Style Guide Team	3 Feb 2014	Initial Public Version (FDI LLC)
1.0.1	Style Guide Team	1 Nov 2014	First Release (FDI LLC)
1.0.1b	FCG	10 Sep 2018	Put 1.0.1 on FCG Doc Template Corrected bad dictionary section references in 10.1. Ready for release in FCG FDI Device/Package Registration Kits

1 Introduction

The core of the FDI technology is the scalable FDI device package. The FDI device package is a collection of files: The Electronic Device Description (EDD) includes the device definition (DD), business logic (BL) and user interface description (UID). It is based on Electronic Device Description Language (EDDL, IEC 61804-3). The optional user interface plug-in (UIP) offers the advantages of freely programmable user interfaces familiar from FDT, based on Windows Presentation Foundation. Via the FDI device package manufacturers define which data, functions and user interfaces are stored on the FDI server.

The device definition describes the accessible field device data and its internal structure (e.g. blocks). The business logic primarily ensures that the device data remain consistent, e.g. refresh data when unit is changed. The dynamic dependencies, in particular, play a part here, e.g. options and ranges that depend on prior selection of other settings. User interface descriptions and user interface plug-ins define the user interfaces to the field device. Product documentation and protocol-specific files such as device database files (GSD) or capability files (CFF) can be added to the FDI device package as attachments.

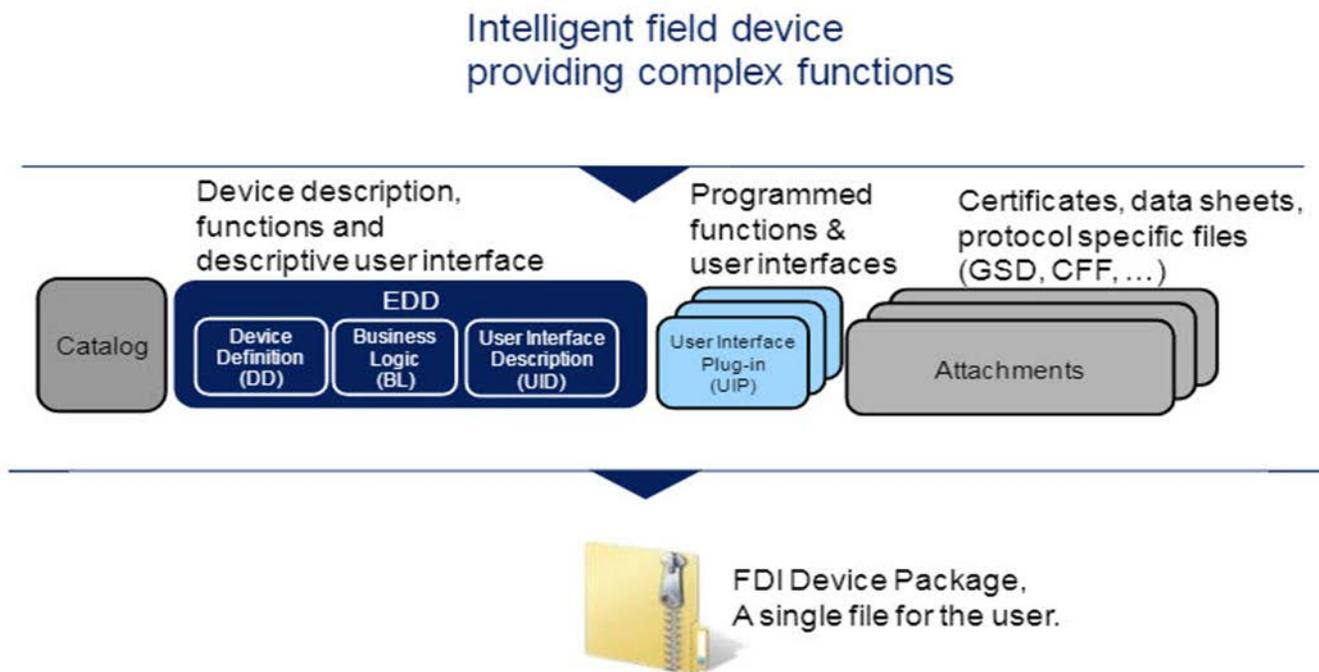


Figure 1 FDI device package (according to white paper 23-Jan-2012)

1.1 Abbreviations

The following abbreviations are used in the Style Guide.

BL	Business Logic
DD	Device Definition
EDD	Electronic Device Description
EDDL	Electronic Device Description Language
FDI	Field Device Integration
FDT	Field Device Tool
GUI	Graphical User Interface
NAMUR	User Association of Automation Technology in Process Industries
OEM	Original Equipment Manufacturer

PNG	Portable Network Graphic
UI	User Interface
UID	User Interface Description
UIP	User Interface Plug-in

1.2 Terms & definitions

1.2.1 FDI specific terms

The Field Device Integration Project, Technical Specification - Part 1: Overview in IEC 62769-1 defines the Terms used in this Style Guide. Other terms and definitions are described as follows.

1.2.2 Data type names and references to data types

The conventions for naming and referencing of data types are explained in EN 62453-2 clause A.1.

1.2.3 Vocabulary for requirements

The vocabulary for requirements is used as defined in ISO/IEC Directives Part 2 (Edition 6.0 2011-04) The following expressions are used when specifying requirements.

Usage of "shall" or "mandatory"	is to is required to it is required that has to only ... is permitted it is necessary
Usage of "should" or "recommended"	it is recommended that ought to
Usage of "may" or "optional"	is permitted is allowed is permissible

1.2.4 Specific formatting

The following formatting is used to describe specific context.

CAPITAL LETTERS	Key-words described in other documents, e.g. SEPARATOR defined in IEC 61804-4
[Button text]	Button with the specified text
<Element name>	Name of an XML element according to data type definition in IEC 62453-41
Code example	A type, class, identifier etc. mentioned in the text.
→UID, →UIP	Denotes a technology specific section

1.3 Normative references

This document is related to and/or references the following documents:

EN 62453-2, "Field Device Tool (FDT) interface specification," 2009.

IEC 61804-3, "Function blocks (FB) for process control and Electronic device description language (EDDL) - Part 3: EDDL syntax and semantics," COMMITTEE DRAFT (CD), 2012.

IEC 61804-4, "EDD interoperability guideline".

IEC 62769-3, "Field Device Integration Project, Technical Specification - Part 3: FDI Server," 2011.

EN 62453-61, "Field device tool interface specification - Device Type Manager (DTM) Styleguide for common object model", 2009.

ISO/IEC Directives Part 2 (Edition 6.0 2011-04)

IEC 62453-41, "Field device tool (FDT) interface specification - Part 41: Object model integration profile - Common object model"

ISO EN 9241-1, "Ergonomic requirements for office work with visual display terminals (VDTs) - Part 1: General Introduction", 2002

ISO EN 9241-14, "Ergonomic requirements for office work with visual display terminals (VDTs) - Part 14: Menu dialogues", 2000

ISO EN 9241-110, "Ergonomics of human-system interaction - Part 110: Dialogue principles", 2006

ISO EN 9241-143, "Ergonomics of human-system interaction – Part 143: Forms", 2012

1.4 Purpose

The purpose of this document is to describe usability related aspects of the design of an FDI device package. This document provides additional information related to the technical specifications given in IEC 62769 and IEC 61804; it does not overrule these standards.

The benefits of good UI design shall be motivation for FDI device package designers of UID and UIP to apply the FDI Usability Style Guide.

Designing highly usable FDI device packages is challenging. First, the FDI device package shall provide an easily understandable and task oriented structure to the large number of parameters and functions of the field device under design. Second, the structure should be consistent with other field devices of the same class or family to allow end users to transfer their handling knowledge between these field devices. And third, the field device will be but one of many devices handled by maintenance personnel and device specialists.

This document supports FDI device package designers to meet the above mentioned challenges for **new** FDI device packages. It matches generic usability guidelines and rules with the FDI specification and shows how to achieve the following goals:

- **Uniform user guidance**
 - Task related structuring of the field device's information in hierarchical menus
 - Fixed structure for the root nodes of the navigation structure

- **Uniform behaviour independent of implementation technology (UID, UIP)**
 - How to display information or use services about the device state
 - Designing parameter groups with resizeability in mind
 - How to use the services to provide uniform buttons
 - Common status handling

1.5 Field of application

The FDI Usability Style Guide is provided for designers of UID and UIP of newly designed FDI device packages. It is assumed that FDI device package designers are familiar with the concepts and services of IEC 62769 and the EDDL specified in IEC 61804.

1.5.1 UID designers

UID designers are responsible for the development of the UID part of an FDI device package which is expressed in EDDL. The UID part defines the navigation structure and the elements of displays in a textual way. UID designers should use tools such as the FDI Package IDE for designing the UID.

1.5.2 UIP designers

UIP designers are responsible for the development of the UIP part of an FDI device package. An UIP is implemented with a graphical UI programming framework. This makes it possible to implement highly interactive and device specific graphical dialogs for complex tasks. UIPs are typically programmed within environments such as Visual Studio.

1.6 How to use this document

The FDI Usability Style Guide gives recommendations for several aspects of UI design for FDI device packages. It is structured as follows: The introduction makes designers familiar with scope, terms and references of this document. Section 3 “Usability goals” provides an introduction to usability as design goal, section 4 “Overview of functionality” general design heuristics are introduced that help to achieve high usability. In section 5 “User roles” designers learn about the context and tasks of their end users. Section 6 “Use cases” gives recommendations when to switch from UID to UIP-development. Section 7 “Platform capabilities and constraints” describes the set of design related features designers can expect of any FDI host. Section 8 “Menus” gives recommendations for the structuring of the interaction space with menu. In section 9 “Dialog design” readers learn how to use containers, controls and other UI elements as well as colors and contrasts. Section 9 addresses the general design aspects of dialogs, section 10 “Parameter handling” gives hints on how to display and edit device parameters.

The FDI Usability Style Guide includes examples of source code or sketches of the graphical representation of controls or frames. These examples serve to give designers an idea how to code or how the code may be rendered in the UI. These examples **shall not** be understood as a general recommendation.

2 Usability goals

Usability is defined as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” (ISO 9241-1). To support FDI device package designers the FDI Usability Style Guide therefore not only describes the users, their goals and their context of use but also gives concrete recommendations on how to ensure effectiveness, efficiency, and satisfaction with the particular technologies and constraints of an FDI device package.

2.1 Specified users and their goals

User roles (section 5) and use cases (section 6) are described to provide FDI device package designers with a basic understanding of the tasks and responsibilities of the end users who will work with the user interfaces.

2.2 Context of use

Many aspects of the context of use of a FDI device package are determined by the FDI hosting environment. FDI package designers need to expect these environments to be as different as mobile handhelds, laptop or tablet based field maintenance stations, or HMI of process control systems in control rooms. Therefore the FDI Usability Style Guide challenges and supports UI designers to generate designs that work in a variety of environments.

2.3 Effectiveness, efficiency and satisfaction

Usability is measured in terms of effectiveness, efficiency and satisfaction. ISO EN 9241-110 provides general user oriented design principles that should be taken into account when designing interactive systems:

- a) Suitability for the task;
- b) Self-descriptiveness;
- c) Conformity with user expectations;
- d) Suitability for learning;
- e) Controllability;
- f) Error tolerance;
- g) Suitability for individualization.

The FDI Usability Style Guide provides measures to implement these principles and to design usable UI whose effectivity and efficiency can be quantified by

- acceptable level of understanding of the structure of displays and menus
- acceptable response and update time of the system
- acceptable navigation efforts and exploration or search times
- acceptable error rates

3 General usability heuristics

Nielson's ten usability heuristics are quite popular rules of thumb for implementing the above mentioned abstract design principles that should be taken into account when designing an FDI device package.

3.1 Visibility of system status

The system should always keep users informed about what is going on, for example through appropriate feedback within an acceptable time limit. Visibility contributes to self-descriptiveness. Important elements of the FDI UIs are icons (e.g. NAMUR icons) or colors that inform about the status of parameters (current, modified, invalid).

3.2 Match between system and the real world

The system should speak the end users' language and use familiar words, phrases and concepts rather than system-oriented terms. Follow real-world conventions, e.g. make information appear in a natural and logical order. This design principle of conformity with user expectations adds to suitability for learning as well. FDI device package designers implement this heuristic with user oriented dictionaries for the labeling of menus, parameters, and tooltips.

3.3 User control and freedom

Users often choose system functions by mistake and need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. While a full multi-stage undo/redo implementation like users experience with their office applications is not always possible with devices that interact with physical processes; at least saving and loading of configurations should be implemented to allow undoing erroneous activities. FDI supports this heuristic by maintaining a dataset of a device with an edit mode and predefined services to apply changes (chapter 10).

3.4 Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Be consistent and follow platform conventions to implement the design principle of conformity with user expectations. The FDI Usability Style Guide provides a dictionary for frequent top-level terms and gives hints on how to name user-defined menu entries.

3.5 Error prevention

A careful design which prevents a problem from occurring in the first place is even better than good error messages. Examples are structured option groups and clearly visible hints of the expected and valid data types, for example by displaying ranges. Error prevention by design may include design principles such as conformity with user expectations, controllability and error tolerance. EDDL provides standardized means to design methods that validate inputs for variables and variable types before sending them to the field device.

3.6 Recognition rather than recall

Make objects, actions, and options visible. End users should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. In short, apply the design principle of self-descriptiveness. FDI package designers implement this heuristic with variable types that have predefined choices, well written labels for variables and option groups, self explanatory wizard structures and logical grouping of device variables in task oriented displays.

3.7 Flexibility and efficiency of use

Accelerators - unseen by the novice user - may often speed up the interaction for the expert user so that the system may cater to both inexperienced and experienced users. This may be supported by EDDL's capability to define keyboard shortcuts for single items. However as in the previous heuristic, a logical task- oriented grouping of parameters contributes even more to efficient work flows.

3.8 Aesthetic and minimalist design

Displays should not contain information which is irrelevant or rarely needed. Every extra unit of information in a display diminishes the relative visibility. Less is more!

3.9 Help users to recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

3.10 Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focus on the user's task, list concrete steps to be carried out, and not be too large. In FDI in particular, well written tooltips contribute to implementing this heuristic.

4 User views

A user role specifies the set of tasks that need to be performed by this role. FDI supports roles by the concept of customer adaptable user views. The FDI user view model distinguishes the basic user views *Maintenance* and *Specialist*.

4.1 Maintenance

The user view *Maintenance* covers all core functions and variables/parameters for the commissioning of a new device which has been delivered by the manufacturer with default parameters and the exchange of a device. This includes device commissioning, function test, device calibration, condition monitoring, data synchronization and device replacement (see section 6).

This view should be implemented with UID elements only. Functions and Variables that should not be included into the user view *Maintenance* should be marked with the CLASS Attribute value SPECIALIST. The default definition of the core variable/parameter attributes for *Maintenance* shall be provided by the manufacturer based on general customer requirements (e.g. NAMUR recommendations).

An FDI host may provide tools that make it possible to override the default mapping and adapt the predefined settings to the needs of a specific customer.

4.2 Specialist

The user view *Specialist* provides unrestricted access to all device functions and variables, which are marked with the CLASS attribute SPECIALIST, including the core variables from the user view *Maintenance* (see Figure 2).

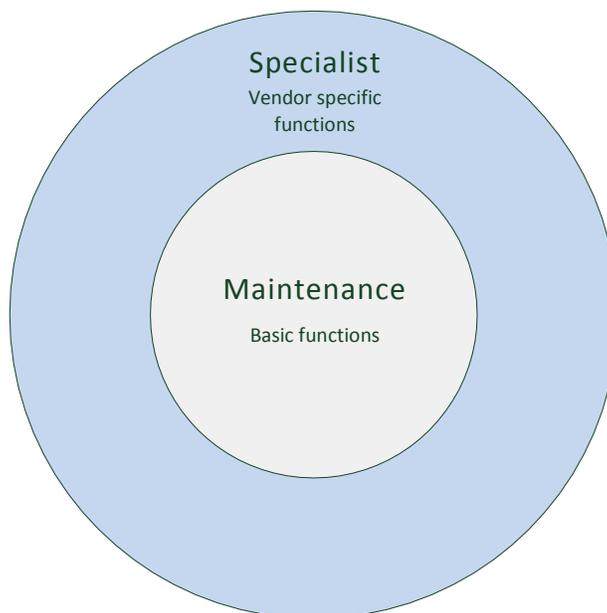


Figure 2 FDI user view model

5 Use cases

"A use case *specifies* a closed set of *tasks* provided by a *system* to provide *recognizable benefits* to the actors." [1, p. 201] The use cases defined in this section belong to the user role *Maintenance* (see section 5). The user role *Specialist* shall be able to perform all these tasks.

5.1 Device commissioning

The user deploys the device application database to a device and uploads device specific data into the device database for archiving. The user validates the application data sent to the device and ensures that the device is operational and configured for proper process operation.

5.2 Function test

The user verifies that the device is operating correctly after deployment of device data from the engineering database. The user verifies that the device calibration is appropriate for the intended process application and that the device is consistently configured. The user executes diagnostic functions and modifies the device configuration as needed to ensure the correct behavior of the device.

5.3 Device calibration

The device is adapted to the specific physical properties of some of its elements (sensors), possibly in conjunction with its environment (measurement target).

5.4 Condition monitoring

The information available on system entities is analyzed to identify predefined system conditions (device failures, location of a newly commissioned device, etc.). For well-known conditions, their occurrence may already imply the corresponding root-cause. In all other cases, an additional fault diagnosis is required to deliver input data to the root-cause analysis.

5.5 Data synchronization

The user reconciles the online data in the device against the data stored for the device in the engineering database. The user validates the device configuration as defined in the engineering database against the configuration currently present in the device. The user reconciles differences generating a consistent data set between the device and the device database.

5.6 Device replacement

A device in the process is replaced with an identical device, a device at a different revision level, or a similar device. A device repair that results in a change of the configuration of the field device belongs to this use case as well.

5.7 Device decommissioning

The user prepares a device for removal from the process. The current device's data is synchronized into the device database to both document the system as last executed as well as for re-use of the device configuration in future system environments.

6 Platform capabilities and constraints

The FDI host manages both UID hosting and UIP hosting. From a usability perspective it is important to ensure that there is no rupture in user experience when switching between technologies. The FDI host supports this with architectural features such as providing a set of standard UI Actions which help to implement UI designs that seamlessly integrate in different environments.

6.1 Hosting

The appearance and layout of a particular FDI host can be quite specific and will vary from product to product. Figure 3 shows different layouts. The left and middle layouts show wire frames of an FDI host implementation that utilizes menus as main interaction metaphor to maximize the space available for the application area, the right layout uses display areas with the elements always directly accessible.

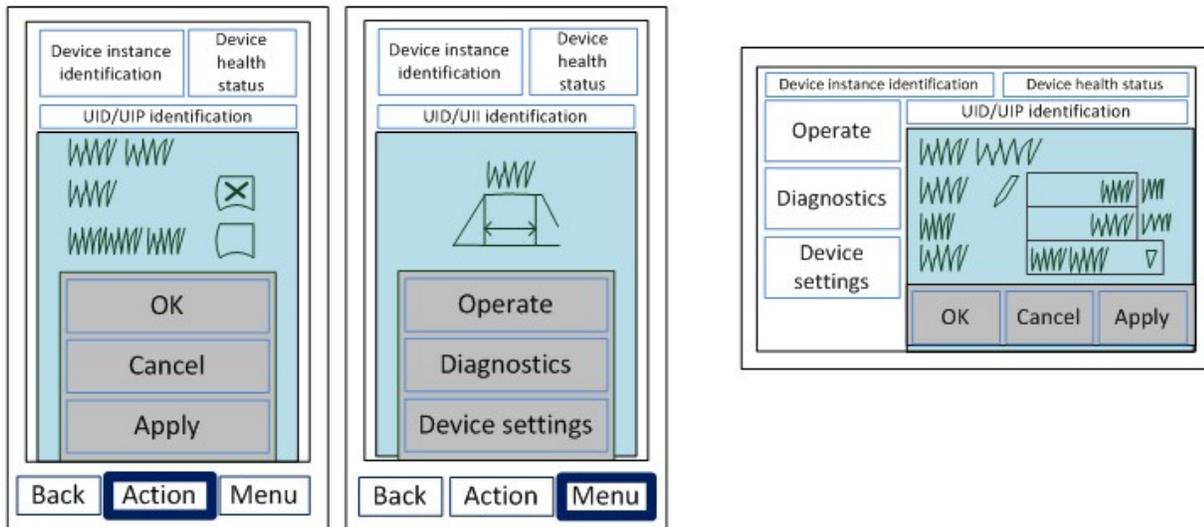


Figure 3 - Example wire frame that illustrate the expected diversity of screen layouts of FDI hosts.
Left: Action element realized as menu.
Middle: Navigation element realized as menu.
Right: Action element and navigation element realized as display areas.

FDI device package designers **directly** influence the following **three** aspects of the appearance of the FDI device package in the context of the FDI host:

- Content of the navigation element - by means of the menu structure given in the UID.
- Content of the application area - by means of the displays defined as UID or UIP (blue colored)
- Content of the action element - by means of predefined services of the FDI host

An FDI device package designer may **indirectly** influence the information shown in the identification element of the FDI host. The FDI host may select relevant information from the FDI package catalog, such as the device picture or the device instance identification. Details about the elements of the FDI package catalog are provided in IEC 62769-4.

6.1.1 Navigation element

Every FDI host will provide an interface element that allows end users to access at least the top-level menus (see section 8.1), for instance a drop-down menu (Figure 3 middle) or an explorer view (Figure 3 right). Whenever the end user selects and activates one of the entries in the navigation element the FDI

host will display the related UID or UIP in the application area and its corresponding action elements in the action element.

6.1.2 Action element

The FDI host provides an action element that holds the user interface elements for terminal actions on UID and UIP alike, that is apply changes and close (UI Action OK), cancel changes and close (UI Action CANCEL), or apply changes and keep dialog open (UI Action APPLY). As depicted in Figure 4 an FDI host implementation may implement those elements with the prevalent interaction technologies and the look and feel of the host's operating system.

6.1.3 Automated generation of action elements →UID

Within a UID an FDI device package designer cannot and needs not influence the elements of the action element. The FDI host automatically adds OK, CANCEL, and APPLY UI Actions.

6.1.4 Services for StandardUIActions →UIP

UIP designers shall use the `GetStandardUIActionItems` service (IEC 62769-2) to adapt the action element to the needs of the UIP dialog. The parameters of this service make it possible to enable and disable the standard UI Actions OK, CANCEL, and APPLY.

For some UIPs it may be helpful to add one or two user defined actions to the action element. It is recommended to use this feature sparsely and only when clearly recognizable by end users. To utilize this feature UIP designers define `SpecificUIActionItems` and implement the service `GetSpecificUIActionItems` (IEC 62769-2).

6.2 Input and output devices

6.2.1 Keyboard

FDI package designers can expect that all platforms are equipped with a keyboard, physically or virtually.

6.2.2 Pointing devices (mouse or touch screen)

FDI package designers can expect that most platforms are equipped with a pointing device – either physically (mouse, touch screen) or emulated (for instance cursor and arrow keys). In particular UIP shall therefore be prepared to be operated by keyboard only.

6.2.3 Screen

FDI package designers shall expect that the screen sizes (pixel, aspect ratio) and screen capabilities (color, resolution) of the target platforms vary widely. Designing UID and UIP for a fixed screen size is therefore not recommended (see section 9.4).

6.2.4 Printer

The FDI host provides capabilities to print the parameter set of a device described in the UID.

7 Top-level menus

The UID of an FDI device package shall provide a fixed set of predefined main entry points, the top-level menus. These top-level menus provide end users with a stable set of access points to the various functions of a field device via the navigation element of the FDI host. All further structuring of the user interface of the field device (page, window, group) shall be accessible as items of the top-level menus.

7.1 Mandatory top-level menu entries →UID

The UID of a new FDI device package shall contain at least one of the mandatory top-level menus (Table 2). It is recommended that UID designers implement the top-level menus in the order given in Table 2 to ensure a consistent and stable end user experience independent from the FDI hosts capability to resort menus.

Identifier of the top-level menu	English default label	FDI user view Maintenance	FDI user view Specialist
process_variables_root_menu	Operate	subset of variables	all variables
diagnostic_root_menu	Diagnostics	subset of variables	all variables
device_root_menu	Device settings	subset of variables	all variables

Table 2 Top-level menus

The subset of variables in the FDI user view Maintenance is determined by filtering those variables that are marked with the variable class attributes DEVICE_COMMISSIONING or DEVICE_MAINTENANCE.

7.1.1 Operate

The Operate menu includes displays that show process measurements and set points with their quality as well as important information for process operators, e.g. ranges and boundaries. Furthermore, it is good practice to provide displays that make it possible to change operation related variables, such as set points or alarm boundaries. Ideally, all the operation-related tasks could be executed with the displays under this top-level menu.

7.1.2 Diagnostics

The Diagnostics menu includes displays that show the device state, detailed diagnostic information and may include graphical presentations that show, for example showing a valve signature. Whenever it is necessary to set device parameters for running diagnostics or filtering results it is recommended to add these parameters under this menu to achieve efficient diagnostics workflows. Otherwise end users would be forced to switch between top-level menus within one task.

7.1.3 Device settings

The Device settings menu includes features for device support. It is recommended collecting all parameters under this top-level menu which are not directly modified in standard diagnostics or operation tasks.

7.2 Structures below the top-level menus →UID

7.2.1 Usability aspects of the menu structure

There are often too many variables for three single displays as implementation of the top-level menus. FDI UI designers therefore need to provide an easily understandable and navigable hierarchy of displays.

It is recommended to use not more than eight items per menu node to support mobile devices as well as desktop applications and to implement the following structural pattern below the top-level menu.

- [MENU]+ →WINDOW →[PAGE] →[GROUP] →Variable, Method etc.
- [MENU]+ →DIALOG →[PAGE] →[GROUP] →Variable, Method etc.

Square brackets ([]) denote optional elements; a plus sign (+) denotes possible hierarchical structures.

If the frequency of usage can be estimated or determined then the most frequently used options should appear first.

Every MENU object shall have a LABEL attribute which provides the visible text for the end user. The FDI device package designers follow the following recommendations:

- Labels should be short, meaningful and self-explanatory (see tooltips for how to provide further information),
- Labels should be written for end users, not for device developers,
- Labels should be fragments with sentence capitalization

7.2.2 Representation

FDI package designers use EDDL's menu element to implement the navigation and display hierarchy. The STYLE attribute of the menu element defines how the FDI host shall render the information. Table 3 summarizes the recommendations for UID designers.

STYLE	Description	Recommendation
MENU	The items of the GUI should be displayed as a menu.	The number of entries in a menu should be limited to eight items.
WINDOW	Other GUIs can be activated while this GUI is active. Interactions with all activated GUIs having WINDOW STYLE attributes are possible, i.e. specifies that the items of the GUI should be displayed as a modeless window.	
DIALOG	No other GUI can be activated while this GUI, or any of its submenus, are active. The GUI shall be closed before interactions with other GUIs are possible, i.e. specifies that the items of the GUI should be displayed as a modal dialog box.	It is recommended to use DIALOG for wizards and critical sections only (see section 9.8)
PAGE	The items of the GUI item should be displayed in one tabbed property page.	The number of PAGEs per display should not exceed nine
GROUP	The items of the GUI should be displayed as a group box.	The number of GROUPs should not exceed nine
TABLE	The items of the GUI should be displayed in vertical columns.	TABLES may be used as item of a GUI with the STYLE attribute WINDOW or PAGE The number of TABLES should be reduced.

Table 3 The UID STYLE attributes (Columns STYLE and Description according to IEC 61804-4)

8 Dialog design

A general design principle is to implement as much as possible functionality in the UID. UIPs shall be used to support the UID part of an FDI device package.

This section gives recommendations according to ISO 9241-143. This standard describes how to use common UI elements that can be found both in desktop and mobile applications. Several implementation hints and examples are adopted from the EDD Interoperability Guideline IEC 61804-4 for UID designers. As the UIP implementation of these UI elements is technology dependent and therefore subject to frequent change, programming examples for UIP are not in the scope of this style guide. This also the case for user defined UI elements, e.g. user defined controls, as well.

8.1 Fully labelled variables →UIP

An UIP that displays a fully qualified variable shall follow the conventions of the UID. The following basic rules shall be applied to align the textual representation of parameter values

- Textual values should be displayed left-aligned
- Numerical values should be displayed right-aligned.

The elements of a fully labelled variable shall be aligned as follows:

- All label fields are vertically aligned at the left border, the text is left aligned
- All value fields are vertically aligned at the left border and have the same width, the text is aligned according to the rules given above.
- The optional engineering unit field directly follows the value field, the text is left aligned.
- The variable value status icon (see section 10.2) is placed between label and value field.

Parameter a	*	123.45	Unit a
Parameter aa	*	Text aa	
Parameter aaaa	*	12.3	Unit aaaa
Parameter aaaaa	*	Text aaaaa	
Parameter b	*	0.00123	Unit b
Parameter bb	*	Option a	▼

* Placeholder for Parameter State Icon

Figure 4 Variable value and associated information

8.2 Variables with predefined values

Many variables of a field device can take only predefined values. This should be reflected in the user interface by providing the appropriate choices. Two fundamental cases need to be distinguished, mutually exclusive choices (one option out of many) and multiple non-exclusive choices (a group of mostly binary options).

8.2.1 Mutually exclusive choices →UID

Users make a choice from a set of mutually exclusive options; i.e. users can choose only one option from a predefined set. Variables of type ENUMERATED(1) provide by definition a mutually exclusive choice. Typically, FDI hosts implement mutual exclusive choices as a drop down list.

```

/** Mutual Exclusive Choice Example **/
VARIABLE dio_mode
{
    LABEL "Digital IO Mode";
    TYPE ENUMERATED(1) {
        {0, "Single"},
        {1, "Differential"}
    }
}

```

8.2.2 Mutually exclusive choices →UIP

UIP designers can choose to implement mutual exclusive choices, e.g. with a drop down lists or a group of radio buttons.

To optimize the space/information ratio, FDI designers should implement mutually exclusive choices as a group of radio buttons for up to three options. The group should be emphasized with a frame and a described with a label.

For variables with more than three predefined options, a representation of the options as a drop down list is recommended. The text field of the drop down list shall show the current (edited) value of the variable.

8.2.3 Multiple non-exclusive choices →UID

Users make decisions between two clearly opposite choices for each item of a group of options. A typical example is setting the single bits of a variable of type BIT_ENUMERATED or a group of related variables. The different options are collected by a menu element of STYLE Group. Typical FDI hosts will implement multiple non-exclusive choices as an option group of check boxes.

```

VARIABLE options
{
    LABEL "Options";
    TYPE BIT_ENUMERATED
    {
        { 1, "Option a"},
        { 2, "Option b"},
        { 4, "Option c"}
    }
}

MENU options_group
{
    LABEL "Description of Option Group";
    STYLE GROUP;
    ITEMS
    {
        options[1],    // Option a
        options[2],    // Option b
        options[4]     // Option c
    }
}

```

8.2.4 Multiple non-exclusive choices →UIP

The standard control for implementing a multiple non-exclusive choice is a group of check boxes or switches. The check box label indicates the selected state, whereas the meaning of the cleared state must be the unambiguous opposite of the selected state. Consequently, check boxes should be used only to toggle an option on or off or to select or deselect an item. Check boxes may be rendered as illustrated in Figure 5. To support end users the group should be emphasized by a frame and a label.

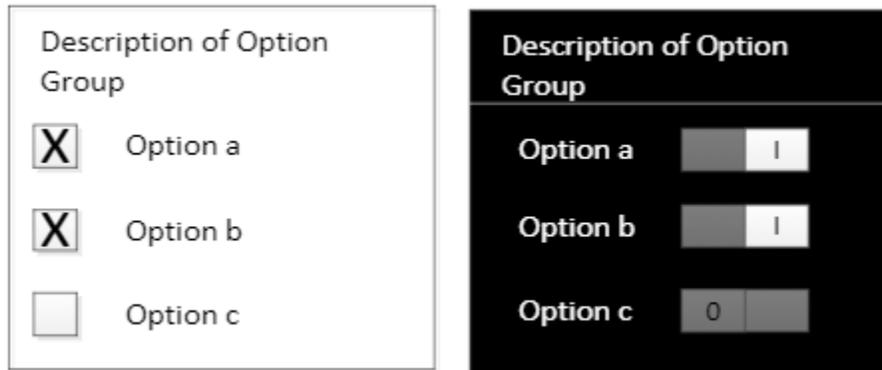


Figure 5 - Wire frame sketches of displays of an option group with multiple non-exclusive choices

8.3 Tooltips

A tooltip is a text that provides further information about an interface element, such as buttons, labels, images or variables. The display of tooltips is evoked by gestures which are specific to the operating system, for example mouse-over and stay or long touch.

The purpose of a tooltip is to inform or instruct the user about the feature and to give a quick guideline on how to use it. They are concise descriptions of a control that provide more information than the label.

Towards that end, a tooltip may provide further information about the type of a variable, its use or the consequences of changing it, as well as information about range restrictions, defaults, or dependencies to other parameters.

The following recommendations should be applied when writing tool tips:

- Use the fewest words possible (minimum two words, maximum 2 sentences)
- Use fragments with sentence capitalization
- Use simple present tense

8.3.1 Tooltips →UID

The FDI host generates tooltips from the HELP Attribute of the interface element of the UID which is a static string. Most FDI hosts will truncate the string to a reasonable length when rendering it as a tooltip. Therefore it is recommended composing a short sentence with up to 255 characters.

Paragraphs (`\n`) should not be used to define beautifying line breaks (e.g. to enforce a certain line width). It is up to the FDI host to render the text as a tooltip and wrap the text.

8.3.2 Tooltips →UIP

UIP designers should utilize the tooltip methods of the operating system with the system default time out and gestures to achieve maximum compatibility to the users' experience.

If the UID of an FDI device package supports multiple language tooltips, the UIP should support this as well. UIP designers shall use the FDI service `ActivateServiceParameters` to retrieve the local settings (IEC 62769-2, Section 6.1.1.1.2).

8.4 Design for resizable →UIP

UIPs should be designed and implemented in a resizable way that is compatible to the resize behavior of UID (see 61804-3). As described in section 7.1 the FDI host will start an UIP in the application area or in an independent popup display.

Displays with a large number of elements that do not fit into the application area or the screen shall provide interactive means to pan the display plane, such as scrollbars, page keys, or touch gestures. It is recommended to implement a flexible layout that needs to be panned only in one direction (typically vertically).

Some presentations provide good usability only with a fixed aspect ratio (images, diagrams). It is recommended that UIP designers provide panning capabilities in both directions.

To implement the above mentioned principles of panning and scrolling along axes it is highly recommended to use the standard elements of the target operating systems, for example scrollbars or touch gestures.

8.5 Text fonts →UIP

To gain compatibility with the FDI Host an UIP should use the default text font for dialogs. This text font is defined in the operating system settings.

Underlined text shall be used exclusively for hyper references.

8.6 Contrast of text and images →UIP

People often have difficulty reading information that does not contrast with its background. This can be exacerbated if the person has a color vision deficiency that lowers the contrast even further. Providing a minimum luminance contrast ratio between the text and its background can make the text more readable even if the person does not see the full range of colors. It also works for individuals who see no color.

The visual presentation of text and images of text shall have a contrast ratio of at least 7:1 Large-scale text and images of large-scale text may have a contrast ratio of at least 4.5:1.

8.7 Colors →UIP

A general design principle is to use color schemes rather than individual colors. To gain maximum compatibility with the end users expectations it is recommended to use the predefined color schemes of the environment. If additional color semantics are necessary, e.g. for variable value states that are not covered by the environment, UIP designers shall use color in a consistent way within an FDI device package.

The conventions for coloring hyper references should follow the operating system of the FDI Host.

8.8 Tables and graphs →UID

Graphs are used to show values of variables as a curve, e.g. a calibration curve. As standard UID-graphs provide no means for the manipulation of the depicted information it is recommended to provide a table if the values of the variables shall be editable by the end user.

8.9 Modal and non-modal dialogs

Wizards (see 9.10) and critical sections such as calibration dialogs should be implemented as modal dialogs.

It is recommended to keep the number of modal and non-modal popup displays as low as possible. Modal popup displays may enforce an inconvenient workflow on the user, non-modal popup displays may lead to cluttered screens.

It is recommended to use non-modal displays for progress information. UID designers may use a CHART with a HORIZONTAL_BAR. UIP designers should use the service `ShowProgressBar`.

8.10 Wizards

Wizard support the workflow of tasks the user has to perform. Often such tasks have several steps for several subtasks. A Wizard should have at least three steps. These are classified in three states:

- Step 1 (Start)
- Step 2 to n-1 (Working steps)
- Step n (Summary)

Wizards should provide the following navigation elements (see dictionary in section 11.1.2 for further languages) in the order given in the following list:

- [`<<Back`] Go back one step (if applicable)
- [`Cancel`] Interrupt the sequence and cancel all intermediate changes.
- [`>>Next`] Advance to the next step
- [`Finish`] Finish the wizard sequence, typically instead of [`>>Next`] in the final summary step

Providing information about the sequence and the progress is recommended good practice. Means of implementation are meaningful labels for the steps, e.g. "Prepare for calibration" instead of "Step 2" as well as numbering the actual step and the total number of steps, for instance "Step 2 of 3".

8.10.1 Wizard state machine →UID

The state machine of a wizard's behaviour is defined as a method. An example for a wizard with three steps is given below. The `MenuDisplay`-function opens the defined window with the appropriate buttons to navigate through the sequence of steps. The variable `select` is evaluated to identify the button pressed by the user.

```

/*Three MENUS for Wizard*/
MENU First_Step_Menu {
    LABEL "Step 1 of 3 - Prepare for calibration";STYLE MENU;
    ITEMS{"Prepare device for calibration", "",min_pressure}}
MENU Second_Step_Menu {
    LABEL "Step 2 of 3 - Set calibration parameters";STYLE MENU;
    ITEMS{"Calibration parameters", "",min_pressure}}
MENU Third_Step_Menu {
    LABEL "Step 3 of 3 - Show calibration results";STYLE MENU;
    ITEMS{"Calibration results", "",min_pressure}}

/*Wizard method*/
METHOD wizard_method{
    LABEL "Calibration wizard";
    HELP "Some help text";
    ACCESS OFFLINE;
    DEFINITION {
    long select; long step; long ret;
    step = 1;
    do{
        switch (step){
            case 1:
                ret = MenuDisplay(First_Step_Menu, "Cancel;>> Next",select);
                if (0!=ret) {abort();}
                if (0 == select) {step = 4;} /*--- Cancel ---*/
                    else if (1 == select) {step = 2;} /*--- Next ---*/

                else {abort();}
                break;
            case 2:
                ret = MenuDisplay(Second_Step_Menu, "<< Back;Cancel;>> Next",select);
                if (0!=ret) {abort();}
                if (0 == select) {step = 1; break;} /*--- Back ---*/
                    else if (1 == select) {step = 4;} /*--- Cancel ---*/
                    else if (2 == select) {step = 3;} /*--- Next ---*/

                else {abort();}
                break;
            case 3:
                ret = MenuDisplay(Second_Step_Menu, "<< Back;Cancel;Finish",select);
                if (0!=ret) {abort();}
                if (0 == select) {step = 2; break;} /*--- Back ---*/
                    else if (1 == select) {step = 4;} /*--- Cancel ---*/
                    else if (2 == select) {step = 4;} /*--- Finish ---*/

                else {abort();}
                break;
        }
        break;
    }
    } while (step < 4);
}
}

```

8.10.2 Wizard implementation →UIP

While UIP designers have much larger degrees of freedom in terms of implementing wizards, they shall use the labels provided in the dictionary (appendix 11.1.2) and stick to the button order given in the previous section.

9 Parameter handling

This chapter describes how data and parameters are handled between the UI display and the device. The user enters or modifies parameters in the UI and transfers the data to the database or to the device. The FDI Usability Style Guide does not differentiate between online and offline modes for data and parameter handling. The same labels shall appear as far as possible for online and offline usage.

9.1 Workflows and caches

The FDI specification eases the design of UID and UIP by providing an architectural model with caches – the **device cache** (device or/and database) that holds data obtained directly from the device or the database and the **edit cache** (UI) that holds changes that result from user interaction. The following chapter do not refer to specific implementation, but is supposed to describe the workflow.

9.1.1 Activate UI

Whenever a display is activated, the relevant data will be loaded from the **device cache** to the **edit cache**.

Menu Pre-Read/Edit and Post-Read/Edit actions are executed.

9.1.2 Edit variables

All the changes due to user interactions will influence the **edit cache** only.

Variable Pre-Read/Edit and Post-Read/Edit actions are executed in this phase.

9.1.3 Update device cache

The modified and validated data will be transferred from the **edit cache** to the **device cache** whenever the user activates the UI Actions [Apply] or [OK].

Post-Write actions are executed in this phase.

9.1.4 Transferring data from/to the device

Transferring data from the device to the database and transferring data from the database to the device is managed by the FDI host and is therefore outside the scope of the FDI Usability Style Guide. This ensures a common behavior that is completely independent from the FDI device package implementation.

9.2 Variable value states

FDI defines four possible states for the display of the value of a variable (Table 4).

State	Foreground Color	Background Color	Icon
Unchanged: The variable's value in the edit cache is the same as in the device cache.	<host>	<host>	
Locally changed value: The variable's value in the edit cache differs from the value in the device cache.	<host>	<host>	Pencil
Invalid value: The variable's value in the edit cache is not plausible, that is one of the validation tests has failed.	<host>	<host>	Exclamation mark
Uncertain value: Whenever there is a possibility that the displayed value of the variable does not reflect the true value, the display shall be marked as uncertain, for instance if an update is missing or the parameter not yet loaded	<host>	<host>	Question mark

Table 4 Parameter States

<host> in this context this means the color defined in the FDI host's color scheme, see section 9.7

9.2.1 Evaluation of variable value state →UID

UID designers provide parameters that describe the validity of a variable's values (MIN_VALUE and MAX_VALUE for numbers, valid choices for ENUMERATED). The evaluation of the variable value state is in the responsibility to the FDI host.

9.2.2 Evaluation of variable value state →UIP

The determination of the variable value state is in the responsibility of the UIP designer. The user input shall be checked as soon as possible. The validation should include checks similar to those recommended to the FDI host (IEC 61804-5):

- Arithmetic variables have a value that lies within the minimum and maximum values
- Enumerated variables have a value that matches one of the enumerators
- All of the bits that are set in a bit-enumerated variable match one of the bit-enumerators
- Index variables have a value that matches one of the indices of the associated reference array

Inadmissible or wrong data that have been recognized should be displayed in UIP with the same text symbols as defined in IEC 61804-5 (see Table 5). These text-symbols shall only be used to display information; users should never need to enter those values (for instance as special values for parameters that would disable options).

Variable Status	Text-Symbol	Alternative	Note
Not a number	-/-	NaN	
Plus infinite	Infinity		result of calculation, e.g. division by zero
Minus infinite	-Infinity		result of calculation, e.g. division by zero
Invalid value	!		the "!" should be in front (left) of the value

***Table 5 Text symbols for inadmissible or wrong data
(IEC 61804-5)***

10 Appendix

10.1 Dictionary

The dictionary sections 10.1.2 and 10.1.3 are compatible with the dictionary of standard terms provided by the FDT Style Guide Version 2.0

10.1.1 Menu labels

Menu identifier	Language	Label
process_variables_root_menu	EN	Operate
	DE	Bedienen
	FR	Manoeuvrer
	ES	Manejar
	IT	Eseguì
	CN	过程变量
	PT-BR	Operar
	JP	プロセス変数
	RU	Обслуживать
diagnostic_root_menu	EN	Diagnostics
	DE	Diagnose
	FR	Diagnostic
	ES	Diagnóstico
	IT	Diagnostica
	CN	诊断
	PT-BR	Diagnóstico
	JP	診断
	RU	Диагностика
device_root_menu	EN	Device settings
	DE	Geräteeinstellungen
	FR	Parametres
	ES	Configuración del dispositivo
	IT	Impostazioni dispositivo
	CN	参数
	PT-BR	Configurações
	JP	機器設定
	RU	Установки прибора

10.1.2 Action area labels

Action identifier	Language	Label
Apply UI Action	EN	Apply
	DE	Übernehmen
	FR	Appliquer
	ES	Aceptar
	IT	Applica
	CN	应用
	PT-BR	Aplicar

	JP	適用
	RU	Применить
OK UI Action	EN,DE,FR, ES,IT,JP,RU, PT-BR	OK
	CN	确认
Cancel UI Action	EN	Cancel
	DE	Abbrechen
	FR	Annuler
	ES	Cancelar
	IT	Annulla
	CN	取消
	PT-BR	Cancelar
	JP	キャンセル
	RU	Отмена
Next Step	EN	Next
	DE	Weiter
	FR	Suivant
	ES	Siguiente
	IT	Avanti
	CN	前进
	PT-BR	Próximo
	JP	次へ
	RU	Далее
Previous Step	EN	Back
	DE	Zurück
	FR	Retour
	ES	Retorno
	IT	Indietro
	CN	后退
	PT-BR	Voltar
	JP	戻る
	RU	Назад
Last Step	EN	Finish
	DE	Fertig stellen
	FR	Terminer
	ES	Terminar
	IT	Fine
	CN	结束
	PT-BR	Terminar
	JP	完了
	RU	Завершить

10.1.3 Variable value states

Variable value state	Language	Label
Locally changed value	EN	Modified
	DE	Geändert
	FR	Modifié
	ES	Modificado
	IT	Modificato
	CN	変更
	PT-BR	Modificado
	JP	変更されたパラメータ
	RU	Изменено
Invalid value	EN	Invalid value
	DE	Unzulässiger Wert
	FR	Valeur invalide
	ES	Valor inválido
	IT	Valore non valido
	CN	无效值
	PT-BR	não-válida
	JP	不正値
	RU	Недопустимое значение
Insecure value	EN	Insecure value
	DE	Unsicherer Wert
	FR	Valeur incertaine
	ES	Valor inseguro
	IT	Valore incerto
	CN	不确定值
	PT-BR	Valor inseguro
	JP	未確定値
	RU	Ненадежное значение

11 Bibliography

- [1] C. Kecher, UML 2 Das umfassende Handbuch [UML 2 A comprehensive manual] 3rd Edition, Bonn: Galileo Press, 2009.
- [2] D. J. Mayhew, The Usability Engineering Lifecycle, a practitioner's handbook for interface design, London: Academic Press, 1999.
- [3] FDI Cooperation, "White Paper, FDI Field Device Integration Technology," 2012.
- [4] B. Shneiderman, Designing the User Interface - Strategies for Effective Human-Computer Interaction, Massachusetts: Addison-Wesley, 1992.
- [5] VDI/VDE 2187, "Uniform user interface for digital field instruments," Beuth, Berlin, 2002.
- [6] A. Geven, R. Sefelin and M. Tscheligi, "Depth and Breadth away from the Desktop – the Optimal Information Hierarchy for Mobile Use," MobileHCI '06 Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, pp. 157-164, 12-15 September 2006.
- [7] S. Bay, Komplexe Menüs und kleine Displays Experimentelle Untersuchungen zur Evaluation und Optimierung der Dialoggestaltungen an Mobiltelefonen, Berlin: dissertation.de - Verlag im Internet GmbH, 2006.
- [8] Microsoft, Windows User Experience Interaction Guidelines.
- [9] W3C, "w3c.org," [Online]. Available: <http://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast7.html>. [Accessed 24 January 2013].
- [10] S. Hooper and E. Berkman, Designing Mobile Interfaces, Sebastopol, California: O'Reilly Media, 2012.
- [11] M. Riedl and F. Naumann, EDDL Electronic Device Description Language, München: Oldenbourg Industrieverlag GmbH, 2011.