# IEC/TR 62453-61

Edition 1.0   2009-08

# TECHNICAL
# REPORT

colour
inside

**Field device tool (FDT) interface specification –
Part 61: Device Type Manager (DTM) Styleguide for common object model**

## About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

# IEC/TR 62453-61

Edition 1.0 2009-08

# TECHNICAL REPORT

**Field device tool (FDT) interface specification –**
**Part 61: Device Type Manager (DTM) Styleguide for common object model**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE      V

® Registered trademark of the International Electrotechnical Commission

# CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

——————————

## FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

## Part 61: Device Type Manager (DTM) Styleguide
## for common object model

## FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC/TR 62453-61, which is a technical report, has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation:

This part, in conjunction with the other parts of the first edition of the IEC 62453 series cancels and replaces IEC/PAS 62453-1, IEC/PAS 62453-2, IEC/PAS 62453-3, IEC/PAS 62453-4 and IEC/PAS 62453-5 published in 2006, and constitutes a technical revision.

The text of this technical report is based on the following documents:

| Enquiry draft | Report on voting |
|---|---|
| 65E/72/DTR | 65E/121/RVC |

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The list of all parts of the IEC 62453 series, under the general title *Field Device Tool (FDT) interface specification*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

---

**IMPORTANT – The "colour inside" logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.**

## INTRODUCTION

This technical report is a user interface design specification for developers of FDT (Field Device Tool) components for Function Control and Data Access within a Client/Server architecture. The technical report is a result of an analysis and design process to develop standard interfaces to facilitate the development of components by multiple vendors that shall interoperate seamlessly.

A device-specific software component, called DTM (Device Type Manager), is supplied by the field device manufacturer with its device. The DTM is integrated into engineering tools via the FDT interfaces defined in this specification. The approach to integration is in general open for all kinds of fieldbusses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

To ensure the consistent management of a plant-wide control and automation technology, it is necessary to fully integrate fieldbusses, devices and sub-systems as a seamless part of a wide range of automation tasks covering the whole automation life-cycle. This integration also requires a consistent look and feel of device specific components.

Figure 1 shows how IEC/TR 62453-61 is aligned in the structure of the IEC 62453 series.
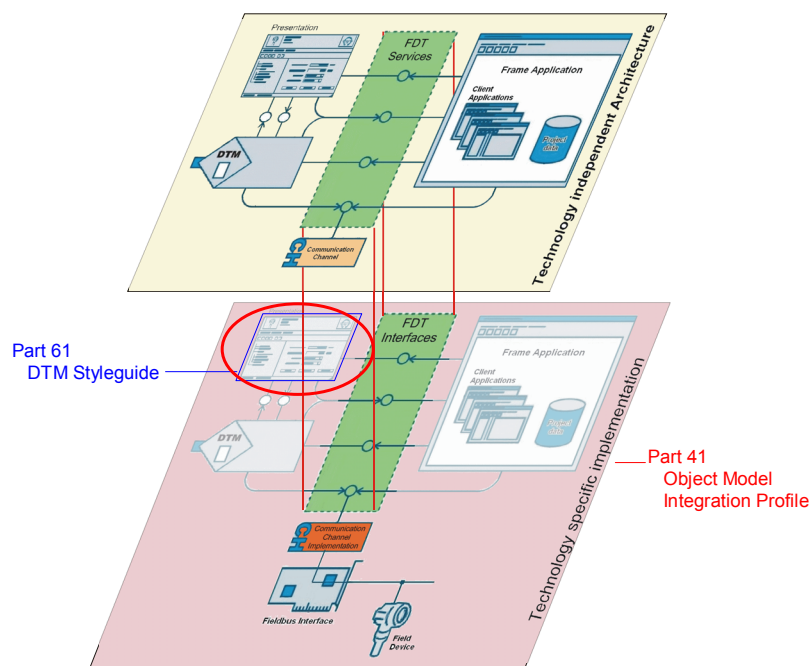


**Figure 1 – Part 61 of the IEC 62453 series**

# FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

## Part 61: Device Type Manager (DTM) Styleguide
## for common object model

## 1  Scope

IEC/TR 62453-61, which is a technical report, explains the guidelines and rules for the implementation of a Device Type Manager (DTM) with regard to the user interface and its functions. These guidelines and rules are part of the FDT specification and are intended to ensure that all users are provided with clear and consistent user interface functions and features across DTM devices in a system.

## 2  Normative references

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

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

IEC/TR 62453-41:2009, *Field Device Tool (FDT) interface specification – Part 41: Object model integration profile – Common object model*

ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML)* Version 1.4.2

## 3  Terms, definitions, symbols, abbreviated terms and conventions

### 3.1  Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62453-1, IEC 62453-2, IEC/TR 62453-41 and the following apply.

**3.1.1**
**screen reader**
software application that provides additional output to users (e.g. text-to-sound, braille)

**3.1.2**
**navigation tree**
GUI element, which displays the navigation information by means of a tree (e.g. tree control)

### 3.2  Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviations given in IEC 62453-1, IEC 62453-2, IEC/TR 62453-41 and the following apply.

GUI          Graphical User Interface

| SUI | Standard User Interface (a GUI layout defined in this document) |
| AUI | Advanced User Interface (a GUI layout defined in this document) |
| CUI | Composition User Interface (a GUI layout defined in this document) |
| MSAA | Microsoft Active Accessibility |
| OEM | Original Equipment Manufacturer |

## 3.3    Conventions

### 3.3.1    Data type names and references to data types

The conventions for naming and referencing of data types are explained in IEC 62453-2, Clause A.1

### 3.3.2    Vocabulary for requirements

The following expressions are used when specifying requirements.

| Usage of "shall" or "mandatory" | No exceptions allowed. |
| Usage of "should" or "recommended" | Strong recommendation. It may make sense in special exceptional cases to differ from the described behavior. |
| Usage of "can' or  "optional' | A DTM may provide the function or behavior depending on the task and type of the DTM. If a function or behavior is provided, it shall follow the style guide. |

### 3.3.3    Specific formatting

The following formatting is used to describe specific context.

| CAPITAL LETTERS | Names of keys on the keyboard —for example, SHIFT, CTRL, or ALT. |
| [Button text] | Button with the specified text. |
| <Element name> | Name of an XML element according to data type definition in IEC/TR 62453-41. |

### 3.3.4    State machine diagrams

Syntax of the state machine diagrams in this document is defined in IEC 62453-1 and in ISO/IEC 19501:2005.

## 4    Principles for designing DTM user interfaces

The design of GUIs for DTMs is based on the following general principles for user interface design [4],[5]. These are recommendations for good engineering practice. For additional fundamentals of user interface design, please see the available literature.

**Visibility of system status**
The system should always keep users informed about what is going on, through appropriate feedback within an acceptable time limit.

**Match between system and the real world**
The system should speak the users' language with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

**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. Support undo and redo.

**Consistency and standards**
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

**Error prevention**
A careful design which prevents a problem from occurring in the first place is even better than good error messages.

**Recognition rather than recall**
Make objects, actions, and options visible. The user 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.

**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 can cater to both inexperienced and experienced users. Allow users tailoring of frequent actions.

**Aesthetic and minimalist design**
Dialogue should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

**Help users 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.

**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.

# 5    Benefits from the FDT user's point of view

Using DTMs compliant with this style guide enables the user to operate more efficiently and more safely. The user is able to parameterize and manage the data of devices from various manufacturers in a uniform way. Therefore, the user is presented with a clearly structured concept regardless of the manufacturer or the type of the device. Details or requirements for developers of a DTM are given within the following clauses.

Guideline and rules are defined for

- uniform user guidance: DTM user interfaces are used and displayed in engineering systems and stand alone tools in the same manner regardless of the device or DTM manufacturer or communication protocol employed;

- uniform behavior of a DTM. This includes:

    - persistent storage,

    - behavior in multi-user environments,

    - error handling;

- clear identification of the DTM and the assigned device;

- ensuring users will be updated on the status and the parameterization of the configuration constantly. All changes of the configuration are marked;

- informing users, whether GUI input affects the device directly or the offline configuration;

- executing plausibility checks of the configuration on a lexical (e.g. only certain characters are accepted), syntactical (e.g. a limited number of characters) and semantically (e.g. given value is below upper limit) correct basis;

- uniform installation/un-installation procedure.

The following screen shots show a Standard User Interface (SUI) (see Figure 2) and an Advanced User Interface (AUI) (Figure 3), two of three possible user interface types (see 7.4.2).



**Figure 2 – Standard User Interface (SUI)**

**Figure 3 – Advanced User Interface (AUI)**

## 6 Functions of a DTM

### 6.1 General

According to IEC/TR 62453-41, a DTM exposes the complete set of available functions with respect to the current state within the XML document returned by IDtm:GetFunctions(). The Frame Application (FA) is responsible for presenting these functions within its overall user interface in a homogeneous way. There should be no break between the DTM based functions and the FDT Frame Application functions.

In general, the Frame Application is responsible for identifying the DTM instance and starting the current function performed by the DTM instance. Such a function can be handled by a specific user interface provided by the DTM. In this case, the Frame Application starts this DTM user interface as integrated application. The Frame Application then shows identification information in a window title bar of this application.

See IEC 62453-2 for a list of predefined applicationIds (Table A.2). Clause 5 in IEC 62453-2 provides information on use cases related to the predefined applicationIds.

### 6.2 Function "Main operation"

A DTM can have a 'main operation' function. This is a special application which aggregates all DTM user interfaces. When 'main operation' is started, it shall show a GUI which identifies the device (ie. user interface of application 'fdtIdentify').

For providing 'main operation' an fdt:StandardFunction entry shall be used with the fdt:applicationId 'fdtMainOperation'. If 'main operation' cannot be assigned to an applicationId, an fdt:Function entry shall be provided (see IEC 62453-2, Table A.13).

## 6.3 Functions "Online Parameterize" and "Offline Parameterize"

A DTM that allows changing the parameter of a device should provide fdt:StandardFunction for parameterization (applicationIds "fdtOfflineParameterize" and "fdtOnlineParameterize").

If a DTM offers user interfaces for configuration or parameterization, it shall be possible to use these applications offline as well as online. This means configuration or parameterization shall be possible without a connected device.

## 7 DTM user interface

### 7.1 Objective

The user interface of a DTM application shall be designed to provide the user with a software component that is easy to use and self-explanatory. The user interface assists the user to be able to concentrate on his main tasks. The user should not be detracted by novel user interface elements or features.

### 7.2 General behavior

#### 7.2.1 General

The user interface of a DTM should be based on the Microsoft Windows Style Guide ([3]). It is recommended to use Windows common controls. Windows common controls shall act in the way defined by Microsoft. That means, it is not allowed to change the behavior of common controls like buttons, combo boxes, edit controls, keyboard shortcuts etc.

#### 7.2.2 GUI navigation

Elements of the DTM user interface shall be selectable by a pointing device (e.g. mouse) and with keyboard. The keyboard shortcuts for navigation among GUI areas and objects shall be supported. TAB-key and SHIFT-TAB combination is used for navigation between Application Area, Action Area and optional Navigation Area. The navigation between objects in Application Area shall be possible also with the same shortcuts. The TAB-order of elements within the Application Area is from upper left to lower right corner, at least for languages that are written from left to right. Middle Eastern languages such as Hebrew and Arabic are written predominantly right-to-left. Consequently, Middle Eastern user interfaces require a different layout. The focus shall change to another area, if the TAB-key is pressed on the last element of the current area. Navigation within a tree view shall be possible with arrow keys.

#### 7.2.3 GUI resizeability

A DTM user interface should be implemented in a resizable way. In this case, the DTM is responsible for supporting re-arrangement of the inner controls. If the initial minimum size is reached, the DTM shall not implement scroll bars because the Frame Application is responsible for that.

#### 7.2.4 Display of information

The display of any information in a visual user interface is accomplished in a textual, symbolic or graphic manner.

Only one font family should be used to optimize the readability of texts (system font is recommended). Only a minimum variation of the font size and font style should be used.

Icons defined in this style guide shall be used only in the defined meaning (see 7.4.3.7.2, 7.4.3.8 and 7.5.3.2). Icons used in addition to the defined icons should be plain and unambiguous and oriented towards existing specifications (e.g. operation system common icons). Tool tips are mandatory for all used icons.

Color shall only be used as secondary information. The color set should be kept small. Flashing information should not be used as a further display attribute. In general, a DTM should use the style chosen within the user settings of the operating system (i.e. Windows).

The currently used color scheme should be explained in the online documentation of the DTM.

All definitions and descriptions inside this Style Guide are related to "left-to-right" languages. Middle Eastern languages such as Hebrew and Arabic are written predominantly right-to-left. Consequently, Middle Eastern user interfaces require an adapted layout.

## 7.3 Microsoft Active Accessibility

Microsoft Active Accessibility (MSAA) [6] is a technology designed to improve the integration of accessibility aids with applications running on Microsoft Windows. MSAA provides a consistent mechanism for exchanging information between applications and assistive technologies. For example, MSAA allows applications to expose screen readers to the type, name, location, and current state of all objects and notifies screen readers of any Windows event that leads to a user interface change.

As a side effect it is possible to use MSAA for fully automated tests of component based software systems. Therefore, a DTM should support MSAA.

User interfaces built with Microsoft common controls typically get full Active Accessibility support without additional development work. However, special attention shall be given if DTM specific user interface controls are used. DTM developers should always verify that these controls are compliant to Active Accessibility requirements.

*Basic Principles of Accessible Design*

Although, no additional development work is required to support Active Accessibility in many cases, the user interface should always be designed with respect to the following five principles:

- support common system size, color, font, and input settings. This provides a consistent user interface across all applications on the user's system;
- ensure compatibility with the High Contrast option. Users desiring a high degree of legibility select the High Contrast option. When this option is selected several restrictions are imposed upon the user interface. For example, only system colors selectable through the Control Panel or colors set by the user may be used by the user interface;
- provide documented keyboard access to all features. This allows the user to interact with the application without requiring a pointing device, such as a mouse;
- provide notification regarding the location of the keyboard focus (i.e. which element receives keyboard input). It should always be apparent to the user and programmatically which part of the user interface has the focus. This requirement also enables use of the Magnifier and Narrator accessibility aids;
- convey no information by sound alone. User interfaces that convey information by sound shall provide additional options to express this information.

## 7.4 Appearance

### 7.4.1 General

Three categories of DTM user interfaces are specified:

- standard layout, where one presentation object displays one application with no or only with limited navigation capability;
- advanced layout, where one presentation object displays one application with advanced navigation capability;

– composite layout, where one presentation object displays multiple applications (for different DTM functions) with advanced navigation capability. The navigation capability also is used for switching between the different applications.

## 7.4.2  DTM user interface categories

In general, a DTM user interface is divided into the following areas:

- Identification Area: contains information about the device that is handled by the DTM;
- Application Area: contains all necessary GUI elements for the selected function;
- Action Area: contains buttons to initiate the user's choice,
- Status Bar: contains global status information about the DTM and device.

These areas shall be arranged as described in the following sections.

### 7.4.2.1  Standard User Interface (SUI)

Figure 4 shows the areas for the SUI of a DTM.

| Identification Area |
|---|
| Menu [1)] |
| Toolbar [1)] |
| Application Area |
| Action Area |
| Status Bar |

**Key**

1) Optional areas

**Figure 4 – Areas of an SUI**

Additional to the mandatory elements (Identification Area, Application Area, Action Area and Status Bar) the GUI may contain a tool bar and/or a menu.

### 7.4.2.2  Advanced User Interface (AUI)

Depending on the complexity of a device and the required functionality, the DTM user interface additionally may provide navigation (see Figure 5).

Such a GUI shall be conform to the AUI layout and provide a Navigation Area.

The Navigation Area provides an overview of the whole parameter set that is related to the current function. In other words, the Navigation Area reflects the data structure. The Navigation Area shall be realized as a navigation tree.

| Identification Area |
| :---: |
| Menu 1) |
| Toolbar 1) |

| Navigation Area | Application Area |
| :---: | :---: |

| Action Area |
| :---: |
| Status Bar |

**Key**

1) Optional areas

**Figure 5 – Areas of an AUI**

Additional to the mandatory elements (Identification Area, Navigation Area, Application Area, Action Area and Status Bar) the GUI may contain a tool bar and/or a menu.

### 7.4.2.3    Composition User Interface (CUI)

The CUI provides access to multiple applications and DTM functions. The user interface is identified with applicationId 'fdtMainOperation' (see 6.2)

The CUI contains the same elements as the AUI.

In contrast to the AUI, the navigation tree shows all integrated applications of the DTM, for example parameterization, diagnosis or device status together with related parameter groups. The left most nodes of the navigation tree shall represent the different DTM applications. All sublevel nodes reflect the structure of data for each application.

### 7.4.3    DTM user interface areas

### 7.4.3.1    General

In general, the user must be aware if modifications of parameters are applied into the data set of the DTM or into the device. Also the behavior of the GUI may vary depending on whether the GUI supports block mode (see 7.5.2.2) or direct mode (see 7.5.2.3).

### 7.4.3.2    Identification Area

This area contains information about the device that is handled by the DTM (see Table 1).

**Table 1 – Contents of Identification Area**

| Height | Width | Contents (read only) | Availability |
|---|---|---|---|
| Maximal 3 text lines | Variable | Device picture (left side) | Recommended |
| | | Company logo of the device manufacturer (right side, optional home page link: use company logo to directly open a web page in an internet browser) | Recommended |
| | | Between picture and logo: | |
| | |    1st line: Device: name attribute of fdt:VersionInformation from DtmDeviceType | Mandatory |
| | |    2nd line: Description: Description according [2] | Recommended |
| | |    3rd line: DTM specific line (Can be fetched from the device, too) | Optional |

### 7.4.3.3    Menu (optional)

This optional area shall be used only in special cases.

The menu shall contain only menu entries which are directly related to the Application Area. That means if application A is opened within the Application Area the menu shall not contain menu entries of application B.

In a CUI menu entries relating to more than one specific application can also be available.

The selection of such a menu item opens the related application within the Application Area.

### 7.4.3.4    Tool bar (optional)

This optional area shall be used only in special cases.

The tool bar shall contain only elements which are directly related to the Application Area.

In CUI additional entries relating to more than one specific application may be available.

Also a [Help] button could be part of the tool bar.

### 7.4.3.5    Navigation Area (conditional)

The Navigation Area is used only in AUI or CUI.

The Navigation Area contains a navigation tree. This tree provides an overview of the whole parameter set. Within the tree the parameter set shall be grouped to appropriate parameter groups.

When selecting a navigation tree entry (leaf or branch) the corresponding parameter group shall be displayed in the Application Area.

The Navigation Area may be hidden or resizable and it can contain scroll bars .

Within a CUI the Navigation Area is also used for selecting different applications. Two approaches are possible:

a)  displaying the applications as left most nodes in the navigation tree (see 7.4.2.3);

b) the Navigation Area can have several tab cards with a tree. Each tab card provides a different view on the same parameters. The tab cards shall be organized according to the users needs. The left most shall be the view relating to applications.

### 7.4.3.6    Application Area

The content of this area depends on the selected application. When the application allows handling of parameters see 7.5 for additional details.

### 7.4.3.7    Action Area

The content of this area depends on the application type as described in the following chapters.

#### 7.4.3.7.1    Application without changeable values

The Action Area contains only a [Close] button. Activation of this button closes the DTM user interface.

#### 7.4.3.7.2    Application with changeable values

The area contains the following buttons in the described order.

[Ok] [Cancel] [Apply]

The button texts shall be adapted to the configured language according to the texts used in standard Microsoft Windows application. The [Ok] and [Apply] buttons will be applied regarding the data source which currently is shown in the status bar (see Table 2).

**Table 2 – Contents of Action Area**

| Button | Data source | Behaviour |
|---|---|---|
| [Ok] | 🗄 | Values changed in user interface will be applied on the instance data set, only. The user interface will be closed (e.g.offline parameterize) |
| [Ok] | 🔧 | All changed values will be transmitted to the device, only. The attribute <modifiedInDevice> (DtmParameterSchema) shall be set. The user interface will be closed (e.g.online parameterize) |
| [Ok] | 🔧 🗄 | Applications contain values from different data sources. Each value of the application itself shall be uniquely assigned to one data source by an appropriate icon (see Table 5). Based on this icon a changed value will be applied on the instance data set or will be transmitted to the device.The user interface will be closed (e.g. online compare) |
| [Cancel] | All states | In general, changed values will NOT be applied. Application will be closed. Please see also 7.2 |
| [Apply] | 🗄 | Same behavior as [Ok], but application will NOT be closed |
| [Apply] | 🔧 | Same behavior as [Ok], but application will NOT be closed |
| [Apply] | 🔧 🗄 | Same behavior as [Ok], but application will NOT be closed |

In a CUI it may be necessary to change the contents of the action area, according to the selected application. In case of switching from applications with changeable values to an application with non-changeable values the DTM shall behave as defined in 9.1.

### 7.4.3.8    Status Bar

The Status Bar contains global status information about the DTM and the device.

Elements of the Status Bar are listed in the following table (Table 3) and are displayed from left to right.

**Table 3 – Contents of Status Bar**

| Height | Width | Contents (read only) | Icon | Availability |
|---|---|---|---|---|
| 1 text line | Variable | 1. DTM connection state | See Table 4 – Possible connection states | Mandatory |
| | | 2. Communication in progress | ↻ | Mandatory |
| | | 3. Data Source | See Table 5 – Possible data source states (for icon and text) | Mandatory |
| | | 4. State of instance data set (AllDataLoaded, validModified, invalidModified) | See Table 6 – Possible states of the instance data set (for Icon and text) | Mandatory |
| | | 5. Changes directly made on the device. Changes have only an impact on the device and not on the instance data set (see use case Online Parameterization). Instance data set and device may not be consistent any longer.<br><br>Status of attribute 'modifiedInDevice' shall be shown as specified in IEC/TR 62453-41. This display shall be provided even if the attribute is not supported in the parameter document of the DTM | (combined icon 'valid modified' and 'device') | Mandatory |
| | | 6. Direct mode active | | Mandatory |
| | | 7. Device diagnosis status. according NAMUR, see [1] (icons with tool tips) | See Table 7 – Possible device diagnostic states (see [1]) | Optional |
| | | 8. OEM Login or User role (refer also to Table 12: relation between user roles and parameter classes) | 'No user logged in' or name of the user who is logged in | Optional |
| | | 9. Progress bar | | Optional |

Possible connection states are given in Table 4 below.

**Table 4 – Possible connection states**

| Icon | Text (English) | Meaning | DTM state | Availability |
|---|---|---|---|---|
| | <no text shall be displayed> | Connecting | going online | Optional |
| | Connected | Connected | online | Mandatory |
| | <no text shall be displayed> | Disconnecting | going offline | Optional |
| | Disconnected | Disconnected | All other states | Mandatory |
| | <no text shall be displayed> | Disconnected - Disturbed | communication set after OnAbort() | Mandatory |
| Empty (no icon) | <no text shall be displayed> | Unknown | going online or going offline | Mandatory<br>In case of optional states are not displayed |

NOTE   The connection state "Disconnected - Disturbed" indicates communication problems (e.g. communication line broken while loading parameters).

Possible Data Source states are given in Table 5 below.

**Table 5 – Possible data source states**

| Icon | Text (English) | Behaviour |
|---|---|---|
| | Data set | Displayed values are loaded from the instance data set. Changed values will be affected on the instance data set only |
| | Data set locked | Displayed values are loaded from the instance data set. Data set is locked |
| | Device | Displayed values are loaded from the device. Changed values will be affected on the device only |
| | Device locked | Displayed values are loaded from the device. Device is locked |
| or<br>or<br>or | Data set / Device | The application contains values from different data sources, for example for the use case online compare (Comparison of offline –from instance data set- parameters with online parameters of the device)<br><br>Each value of the application shall be uniquely assigned to one data source by the appropriate icon |
| No Icon | -/- | Initial data set. This means <fdt:dataSetState> equal to 'default' |

Possible states of instance data set are given in Table 6 below.

**Table 6 – Possible states of the instance data set**

| Icon | State of instance data set |
|---|---|
| empty | allDataLoaded |
| | Valid Modified* |
| | Invalid Modified* |
| empty | Initial data set. This means <fdt:dataSetState> equal to 'default' |
| *Modified parameter value leads to the instance data set state 'validModified' or 'invalidModified', see 7.5.3.2 and IEC 62453–2. | |

Possible device diagnosis states are given in Table 7 below.

**Table 7 – Possible device diagnostic states (see [1])**

| Icon | Device diagnosis states | Explanation |
|---|---|---|
| | Failure | Output signal invalid due to malfunction in the field device or its peripherals. Examples: Sensor break in T transmitters, no echo from a radar L transmitter, defective lamp in a photometer, gas analyzer is not receiving any sample gas flow |
| | Functional check | Output signal temporarily invalid (e.g. frozen) due to on-going work on the device. Examples: Read-out of parameters from a digital field instrument, tank offset L transmitter, calibration of gas analyzer, electrode cleaning of pH measuring instrument |
| | Off-specification | Device is operating outside its specified range or internal diagnosis indicates deviations from measured or set values due to internal problems in the device or process characteristics Example: Bubble formation in flow metering |

| Icon | Device diagnosis states | Explanation |
|---|---|---|
|  | Maintenance required | Although the output signals valid, the wear reserve is nearly exhausted or a function will soon be restricted due to operational conditions. Examples: Weak echo from the radar L transmitter, low lamp intensity in the photometer, consumable reagent in the analyzer is running low |
|  | Output signal valid | |
|  | Diagnosis deactivated | |

## 7.5  Parameter handling

### 7.5.1  Representation within Application Area

If it is necessary to display a large number of parameters in the Application Area, they should be arranged with a tab control or within groups.

A tab control shall contain not more than 5 tab cards in one line.

Grouped parameters shall be displayed with a headline.

### 7.5.2  Change of parameter values

#### 7.5.2.1  Relation between parameters

Change of one parameter value may affect other parameter values. The affected parameters should be adapted to the changed parameter value. For example changing the unit of a parameter could change units of other parameters and the corresponding values are converted according to the new unit. All adapted parameters shall be indicated as changed.

#### 7.5.2.2  Block Mode

In Block Mode users can edit/change one or more parameter value(s). All modified parameter values shall be applied using a button (ie. [Apply] or [OK]) in the Action Area. Therefore, it is possible to modify more than one parameter value before affecting the instance data set or the device data.

#### 7.5.2.3  Direct Mode

In Direct Mode only a single parameter can be applied at a time. Changes have a direct effect on the instance data set or to the device. If a DTM supports Direct Mode, it shall display the respective icon in Status Bar (refer to Table 3). Furthermore it shall indicate the Direct Mode by 'Direct Mode active' in the top most line within the Application Area (not scrollable).

This feature is optional. It should be implemented only if

- a DTM does not support all business rules of a device;
- special functions are supported (e.g. inching mode).

The user must be advised how to store a changed value or to discard changes. This shall be described in the Application Area. In general, the Block Mode is highly recommended.

### 7.5.2.4    Continuous Check and One Time Check

Plausibility checks are performed on a lexical base (e.g. only certain characters are accepted), syntactical base (e.g. a limited number of characters are allowed) and semantical base (e.g. given value shall be below upper limit).

In Continuous Check, while editing parameter values, the plausibility is evaluated after every change of the value, for example after every new character. The following diagram (Figure 6) describes the Continuous Check.
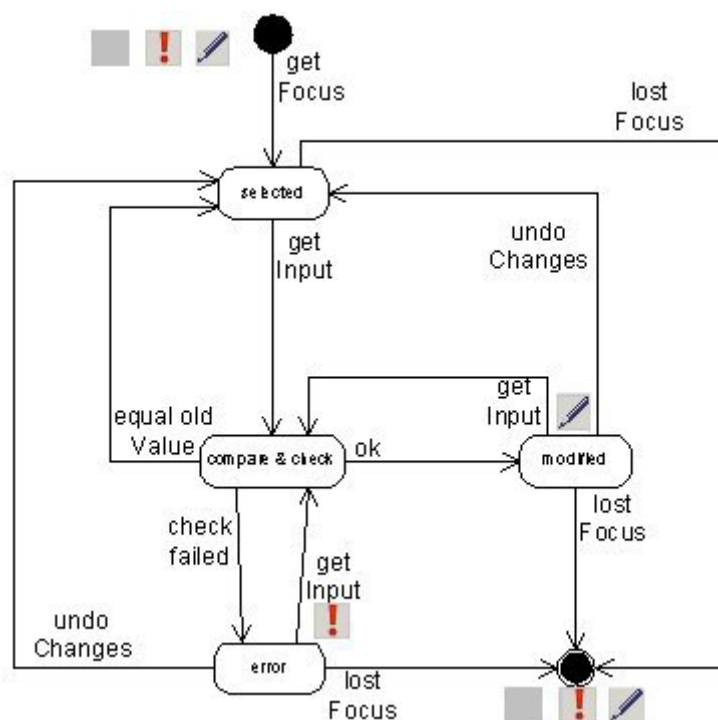


**Figure 6 – State diagram: Continuous Check**

'Lost focus': the input element lost the keyboard focus, for example. if another control of the user interface is selected. Selection can be performed by mouse pointer or TAB.

'Undo changes': the undo function of control is executed (if provided).

Icons are defined in Table 8.

With One Time Check the plausibility of the new value is evaluated when editing of the value is finished. This includes lexical, syntactical and semantical checks.

The following diagram (Figure 7) describes One Time Check.
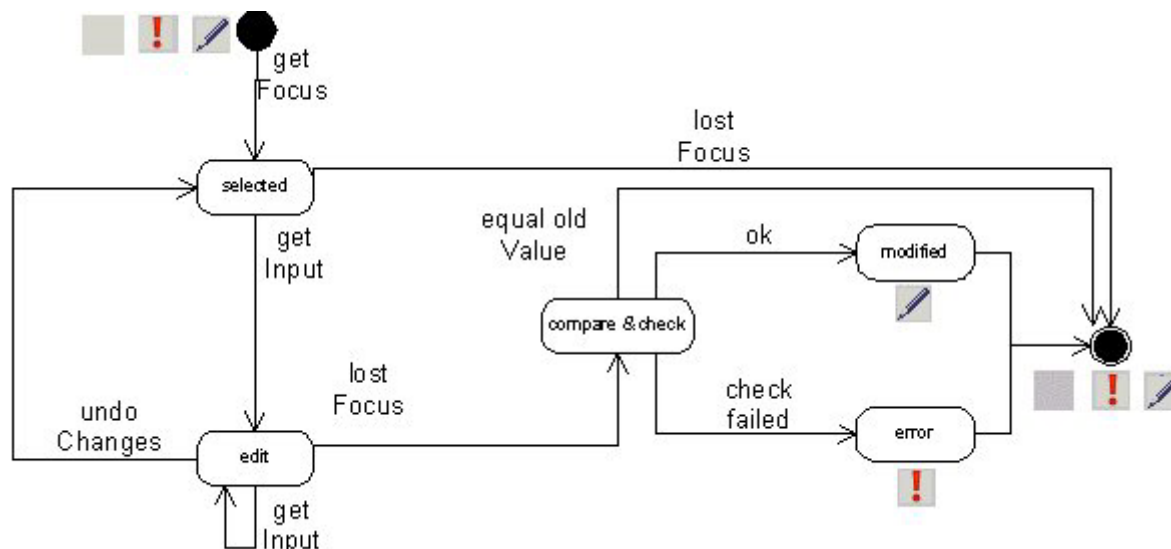
**Figure 7 – State diagram: One Time Check**

In general, Continuous Check should be implemented. If Continuous Check cannot be provided, One Time Check shall be implemented.

### 7.5.3    Representation of parameters

#### 7.5.3.1    Parameter value and associated information

A parameter value shall be displayed with context and status information. The context is represented by a label with the parameter name, a data source icon (when the application displays values from different data sources), a status icon, an input element and a label with a unit (if unit is available).

Example



**Figure 8 – Parameter value and associated information**

- The label for the parameter name should be right-aligned, when arranged in columns.

- The elements for input should be left-aligned.

- The status icon should be in front of the input element.

- If the application displays values from different data sources, the data source icon shall be displayed in front of the input element (between input element and status icon).

- The label for the unit should be left-aligned.

- Inside the input elements, textual parameter values are displayed left-aligned, numerical parameter values are displayed right-aligned.

Where a parameter group displays directly depending parameters (e.g. a group of radio buttons), the group shall display the status information. The status icon should be right from the label for the group.

A short help item (explanation in one sentence, e.g. tool tip) should be available for every parameter, or at least for a group of parameters. A complete help (e.g. via key F1 or hyperlink) for the parameter or parameter group should also be accessible.

### 7.5.3.2 Parameter value modifications

The presentation of each parameter shall identify whether it is modified, not modified or not modifiable regarding the data source. This information is displayed for each parameter independent of the state of the instance data set (transient or persistent) itself. This means that modified parameter values, applied to the data source (e.g. by using an [Apply] button), shall not be marked as changed any longer because the changed values are stored in the data source.

The following Table 8 describes the possible states.

**Table 8 – Possible states of parameters**

| State | Foreground color | Background color | Status flag icon | Text (tool tip) | Availability |
|---|---|---|---|---|---|
| Changeable | --- | <Window> (typically white) | (none) | (none) | Mandatory |
| Not changeable | --- | <3D Objects> (typically gray) | (none) | (none) | Mandatory |
| Not changed Parameter value is equal to data source value (data base or field device) | <default> | --- | (none) | (none) | Mandatory |
| Changed value Parameter value is changed (not equal) to data source | <default> | <Window> (typically white) | ✏ | modified | Mandatory |
| Invalid value (e.g. not plausible) | <default> | <Window> (typically white) | ! | Invalid value | Mandatory |
| Dynamic parameter. Value is updated with polling frequency | <default> | <3D Objects> (typically gray) | 🔃 | (none) | Optional |
| Insecure value parameter is insecure because update is missing for given time period or parameter is not jet loaded from device | <default> | <3D Objects> (typically gray) | ? | Insecure value | Recommended |

NOTE   Color names of the Windows-Desktop are set in <>.
<default> in this context means the color defined in the Windows Color Scheme.
"---" in this context means that the definition of this state is not defined by this color.

Inadmissible or wrong data shall be displayed as shown in Table 9.

**Table 9 – Display of inadmissible or wrong data**

| Data status | Display |
|---|---|
| Not a number |  -/-   or   NaN |
| Plus infinite | >>> |
| Minus infinite | <<< |
| Invalid value | !  234,56 |

### 7.5.3.3    Display of modified parameters in navigation area

Within a navigation tree of the Navigation Area, it must be possible to identify changed or invalid parameters. Therefore, the defined icons for changed or invalid parameters shall be shown at the tree nodes within the tree. It must be possible to navigate from a collapsed tree node to the application containing these changed or invalid parameters. Therefore, every node shall show the status information regarding the states 'Not changed', 'Invalid value' and 'Changed value' by taking also the nodes beneath into account.

## 8   Representation of DTM functions

A DTM can offer several functions which could be available via a menu within the Frame Application. Other options are that the DTM can offer functions within the navigation tree, the tool bar or the menu. The following table (Table 10) shows a list of possible functions of a DTM. Not all functions must be offered by a DTM – each function is optional. If one of these functions is offered by a DTM it shall be available according to the structure defined within the table. If a DTM provides a specific function which could not be mapped to an element of the table this function should be integrated logically from the user's point of view.

Rules for provided functions (returned by IDtm:GetFunctions()):

- all functions provided by IDtm::GetFunctions, shall be independent from a specific user interface. That means functions which can only be executed in the context of a specific DTM user interface shall not be provided by IDtm::GetFunctions, for example Undo, Paste;

- if a provided function implements a behavior which is well-defined within the FDT specification, the associated fdt:StandardFunction entry shall be used. This means:

  - if a function corresponds to an fdt:applicationId, the fdt:applicationId shall be assigned to this function. If a DTM provides only one function for an fdt:applicationId this function shall be defined within the functions document (returned by IDtm:GetFunctions()) as fdt:StandardFunction entry. If a DTM provides more than one function for an fdt:applicationId all these functions shall be defined as fdt:Function entry within one fdt:StandardFunctions entry of the functions document (returned by IDtm:GetFunctions());

  - if a DTM provides a function not assigned to an fdt:applicationId, this function shall be defined as a fdt:Function entry (not with fdt:StandardFunctions entry).

Functions can also be provided within the tool bar according to the rules in 7.4.3.4, and in the menu according to the rules in 7.4.3.3. In this case, the icons defined in the following table (Table 10) shall be used.

All functions shall be organized as defined in the following table (Table 10).

**Table 10 – Representation of functions**

| DTM Function | | | Icon in Tool Bar | <ApplicationId> | Comments |
|---|---|---|---|---|---|
| **1st level** | **2nd level** | **3rd level** | | | |
| Device data | Import{qual}** | | | | Import of DTM specific data e.g. data of a specific format |
| | Export{qual}** | | | | Export of DTM specific data |
| | Page Setup{qual}** | | | | Page setup for DTM specific print function * |
| | Preview{qual}** | |  | | DTM specific print preview* |
| | Print{qual}** | |  | | DTM specific print function* |
| Edit | Undo | |  | | Edit function within a user interface |
| | Redo | |  | | Edit function within a user interface |
| | Cut | |  | | Edit function within a user interface |
| | Copy | |  | | Edit function within a user interface |
| | Paste | |  | | Edit function within a user interface |
| | Find... | |  | | Edit function within a user interface |
| | Replace... | | | | Edit function within a user interface |
| View | Identification | | | | Enable/Disable Identification Area |
| | Navigation | |  | | Enable/Disable Navigation Area |
| | Tool bar | | | | Enable/Disable tool bar |
| | Show all | | | | Show all areas |
| | Advanced Parameter Display | | | | Enables specific additional information in the Application Area based on the current user role |

| DTM Function | | | Icon in Tool Bar | \<ApplicationId\> | Comments |
|---|---|---|---|---|---|
| **1st level** | **2nd level** | **3rd level** | | | |
| Device | Main Operation | | | fdtMainOperation | See IEC 62453–2 |
| | Parametrize Online | | | fdtOnlineParameterize | Changing the device configuration directly in the device. For additional information see IEC 62453–2 |
| | Parametrize | | | fdtOfflineParameterize | Changing the application parameters of a device in the instance data set. For additional information see IEC 62453–2 |
| | Load{qual}** from device | |  | | Transmit a part of the parameter set from the device |
| | Store{qual}** to device | |  | | Transmit a part of the parameter set to the device |
| | Identify | | | fdtIdentify | See IEC 62453–2 |
| | Operate | Observe |  | fdtObserve | Display of actual device information. For additional information see IEC 62453–2 Example: trend visualization |
| | | Trend{qual}** |  | | Trend display of specific parameter |
| | | Adjust set value |  | fdtAdjustSetValue | See IEC 62453–2 |
| | | Audit Trail | | fdtAuditTrail | See IEC 62453–2 |
| | Diagnosis | |  | fdtDiagnosis | See IEC 62453–2 |
| | Simulation | |  | fdtForce | See IEC 62453–2 |
| | Calibration | |  | fdtCalibration | See IEC 62453–2 |
| | Configuration | | | fdtConfiguration | Setting the device configuration. For additional information see IEC 62453–2 Example: change number of modules |
| | Compare online | | | fdtOnlineCompare | See IEC 62453–2 |
| | Compare offline | | | fdtOfflineCompare | See IEC 62453–2 |
| Extras | OEM Login | | | | Start OEM service login |
| | Device reset | |  | | Starts a dialog for resetting the device |
| | Firmware update | | | | Starts a dialog for updating the firmware of the device |
| | Options {qual}** | |  | | Properties for, e.g. the optical representation of a user interface |
| Help | Help for device | |  | | Help for handling of the device |
| | Help for DTM | |  | | Help for handling of the DTM |
| | About | | | | About dialog for the DTM |

a This has no effect on the documentation function offered by the Frame Application (see IEC 62453–2).

b {qual} shall be changed to a meaningful name which describes the DTM specific function in detail.

# 9  DTM behavior

## 9.1  Close of user interface with modified parameter values

When parameter values have been modified within a user interface, and a request is received to close the user interface, the DTM shall ask the user if the changed parameter values should be applied. (e.g. 'Do you want to apply the changes?')

- If 'no' the changed parameter values will NOT be applied and the user interface will be closed.

- If 'yes' the changed parameters will be applied according to the behavior of a [Ok] button or in Direct Mode to the behavior of an [Apply] button.

- If 'cancel' the user interface will not be closed.

This action shall be implemented:

- for a [Cancel] or [Close] button in the Action Area;

- if the Frame Application initiates a close of the user interface (IDtmActiveXControl:PrepareToRelease()). In the case of 'cancel' a DTM shall return FALSE, in all other cases TRUE.

## 9.2  Data set

### 9.2.1  Parameter in multiple user interfaces

When modifying data that is simultaneously displayed in multiple user interfaces of the same DTM instance (same DTM business object) a consistent display shall be ensured by this DTM. The consistent display is triggered at the end of modification of a parameter value in the instance data set.

### 9.2.2  Locking mechanism

#### 9.2.2.1  General

If the data set could not be locked (IFdtContainer:LockDataSet() returns FALSE) or was locked by another DTM (DTM gets notified via IFdtEvent:OnLockDataSet()) data set modification shall be prevented and the corresponding icon within the status bar shall be shown.

Data set modification shall be prevented by disabling all input controls on Application Area or disabling functions to accept the modifications, for example disabling Action Area buttons, Direct mode or menu entries.

A DTM should implement a locking for synchronized DTMs (see IEC 62453–2).

#### 9.2.2.2  Synchronized DTMs

When an event is received regarding changed parameters (IFdtEvent:OnParameterChanged()) the DTM shall take over the new parameter values into the data set. All input controls in all opened user interfaces shall be updated in order to display the new parameter values.

When an event is received regarding an unlock of the data set (IFdtEvent:OnUnlockDataSet()), the DTM shall change the status regarding the locked data set within the status bar (refer to Table 5). If the access rights of the user allow parameter modifications, the DTM shall ask the user if he wants to have write access. The implemented behavior shall not block the IFdtEvent:OnUnlockDataSet() method. When the user wants to have write access, the DTM has to lock the data set. The DTM shall enable the input controls only if the DTM can lock the data set.

### 9.2.2.3  Non-synchronized DTMs

When an event is received regarding an unlock of the data set (IFdtEvent:OnUnlockDataSet()), the DTM shall not enable the input controls and shall not inform the user via a dialog that the data set is unlocked. The DTM shall change the status regarding the locked data set within the status bar (refer to Table 5).

### 9.3  Online parameterization / data source: device

If a DTM user interface provides online access to a device (e.g. for applicationId fdtOnlineParameterization), all displayed parameters shall be loaded from the device and all modified parameters shall be saved into the device without having an impact on the persistent data set. This shall be indicated by setting the appropriate data source icon 'device' in the status bar. Online access to a device shall be provided only if the instance data set is locked by the DTM.

After the user requested to close the DTM user interface, the DTM shall ask the user if the instance data set and the current device configuration should be synchronized by performing an upload operation(the dialog shall provide the buttons [Yes], [No] and [Cancel]). This action may overwrite already existing parameter modifications within the instance data set (status not equal to 'allDataLoaded', refer to IEC 62453–2, 8.7.2). If this is the case, the DTM shall include this information in the message mentioned above.

If the user confirms that the data should be synchronized, the DTM shall upload all parameters into the instance data set. If the DTM can guarantee consistency of device data and instance data set, the DTM is allowed to upload only the modified device parameters under consideration of device specific business rules.

If the user does not confirm the synchronization ([No]), no data shall be loaded from the device.

If the user selects [Cancel] no action shall be performed. The DTM user interface will not be closed.

In general, the instance data set status shall be set as specified in IEC 62453–2, 8.7.2 .

If an upload was aborted due to any reasons, the previous data set and its status shall not change (i.e. same values and status as before upload request).

If the data set is not synchronized by an upload, the DTM shall set the flag 'modifiedInDevice'.

While the parameter values are being loaded from the device (during synchronization), the user interface of the DTM shall provide the possibility to cancel the action. The DTM may not be able to cancel the upload immediately after the user pressed one of the buttons, but it should cancel as soon as possible. The user shall be informed if canceling the action cannot be done immediately.

### 9.4  Offline parameterization / data source: data set

When a GUI of the DTM with applicationId fdtOfflineParameterize is used and when the user requested to close the DTM user interface and the DTM has the possibility to establish a connection to the device, (meaning the DTM is at least in a state of 'communication set'), the DTM shall ask the user if a complete download should be initiated to synchronize the instance data set and the current device configuration (the dialog shall provide the buttons [Yes], [No] and [Cancel]).

If the user confirms the synchronization, the DTM shall download the complete parameter set into the device. If the DTM can guarantee consistency of device data and instance data set, it

is allowed to download only the modified parameters under consideration of device specific business rules.

If the user does not confirm the synchronization (button [No]) no data shall be downloaded to the device.

If the user selects [Cancel] no action shall be performed. The DTM user interface will not be closed.

In general, the instance data set status shall be set according to IEC 62453–2, 8.7.2.

If a download was aborted due to any reasons, the previous data set and its status shall be kept.

## 9.5    Error handling

If an error occurs, the default action shall be safety oriented according to device functionality. For example, if the user has been requested to confirm a device reset, the default answer should be 'No'.

A data set in the state 'invalidModified' shall not be transmitted to the device.

## 9.6    Communication

When the Frame Application is called to set the interface pointer to the communication interface (IDtm:SetCommunication()) the DTM shall not change the internal state to 'online'. Only if a functionality of the DTM requires access to the device, then the DTM shall change the state to 'online'.

After the first online connection to a device is established, a DTM shall check if it can handle this device in a proper way (e.g. by checking the device type via communication protocol specific data). This behavior can lead to two error situations:

* the DTM cannot establish a connection or
* the device cannot be handled by the DTM.

In both error cases, the user shall be informed independently of the function:

* in functions without user interface, an error message shall be given (via IDtmEvents:OnErrorMessage());
* in functions with user interface, in addition to an error message, all controls of the user interface shall be disabled.

If the device cannot be handled by the DTM, the DTM shall go into 'communication set' state.

A DTM which is in 'online' state should go to 'communication set' state as soon as possible in order to support the release of resources. There is no general trigger to go into 'communication set' state. That means 'as soon as possible' is the responsibility of the DTM – the DTM should leave online state as soon as no online functions are necessary.

In the case of FDT related communication errors <fdt:communicationError>, a Device DTM should not repeat the request. Before a Communication DTM generates an FDT related communication error <fdt:communicationError>, it should try to repeat the request independently of the bus protocol.

## 9.7 Access rights

### 9.7.1 FDT actors and parameter classes

The FDT actors and user flags shall be mapped by the DTM for the visibility and for the access to parameters or parameter groups/classes.

Depending on the FDT actors and related user flags, modifications of parameter values may be available or not. If a user is not allowed to modify a parameter, the parameter shall be displayed as read only.

The following table (Table 11) describes the relation between user roles and parameter classes.

**Table 11 – Relation between user roles and parameter classes**

| Actor | Operating-relevant parameters (e.g. measured value, set point value, diagnosis) | Maintenance-relevant parameters (e.g. measuring range, attenuation) | Remaining parameters (e.g. sensor identification data, calibration) | Communication parameters (e.g. PROFIBUS related parameters, addresses, ...) |
|---|---|---|---|---|
| Observer | (read) | (read) | (read) | (read) |
| Operator | read and write | (read) | (read) | (read) |
| Maintenance | read and write | read and write | (read) | (read) |
| Planning Engineer (device specialist) | read and write | read and write | read and (write) | read and write |
| **User Flag** | | | | |
| Administrator (IT-Specialist) | Depends on user role | Depends on user role | Depends on user role | read and write |
| OEM Service | read and write | read and write | read and write | read and write |
| (read), (write): It is not necessary that all parameters of this class shall be provided to this user role | | | | |

In addition to the FDT actors, additional DTM specific user rights can be controlled by DTMs, for example safety devices, and shall be displayed in the status bar of the DTM.

The relation between a specific actor and application is described in IEC 62453–2.

### 9.7.2 OEM login

For security purposes, a timeout for OEM service login shall be implemented by the DTM. The timeout interval starts after each user interaction (e.g. change of parameter value) is finished. The DTM may provide the OEM specific login dialog at the end of the timeout interval again.

Even in OEM login, the DTM shall allow itself to be closed by user interaction or on request by Frame Application (refer to IEC 62453–2, 4.10). Changed parameters will be discarded.

## 9.8 Localization

If a DTM does not support the requested language within the IDtm:SetLanguage() method, it shall use English as the default language.

## 9.9 Documentation

A DTM shall expose all configuration relevant parameters (at least all parameters of application fdtConfiguration and fdtOfflineParameterize) via documentation interface. This means all configuration relevant parameters shall be exposed as <DocumentVariables>.

## 9.10   Installation and un-installation

Before installing a DTM, the setup shall check whether the minimum system requirements are fulfilled. This means at least the operating system and the disk space. Further checks are DTM specific.

Installation of a DTM shall be performed by a wizard that guides the user through several steps as listed in the following table (Table 12). It shall be possible to cancel the installation procedure at any step. In general, the setup procedure should be designed in a way which does not require a reboot. If a reboot is required, it shall be under control of the user to define when the reboot is executed.

**Table 12 – Installation and un-installation**

| Step | Meaning | Availability |
|---|---|---|
| 1 | Selection of user language (English at least) | mandatory |
| 2 | Check if user has Microsoft Windows administrator rights | mandatory |
| 3 | Show name and version of all DTMs which are installable via this setup. If more than one DTM could be installed via the setup, it shall provide means to select the DTMs to install | mandatory |
| 4 | Perform license procedure<br>Accept license agreement | optional |
| 5 | Select if DTM should be installed for all users or only for the current user | optional |
| 6 | Select setup type (basic, professional) | optional |
| 7 | Select installation directory | mandatory |
| 8 | Display all actions that will be performed | mandatory |
| 9 | Perform installation<br>Show errors if any | mandatory |

The default installation directory shall be:

*<drive>:\<program files>\<company name>\...\<DTM related name>*

Depending on the locale of the operation system *<program files>* may vary.

The installation directory or subdirectories shall contain all DTM related data for example ActiveX control, GSD file, XML documents, license file and documentation.

DTM specific registry entries should be contained in the following path:

*HKEY_LOCAL_MACHINE\SOFTWARE\<company name>\...*

User specific registry entries shall be contained in the following path:

*HKEY_CURRENT_USER\SOFTWARE\<company name>\...*

Commonly used libraries should be placed in the *'common files'* directory. If a company specific library is created then a separate company common directory may be needed.

Update procedure shall be supported by the same installation wizard. New releases of a DTM shall be installed easily without manual un-installation of older releases.

Un-installation shall be provided. It removes all parts of a DTM except common used libraries or data (see FDT specification).

A DTM or a set of DTMs shall be uniquely identifiable within the operating system dialog 'Add or Remove Programs'. That means support information (version number/build index) shall be available to identify the version of the DTM.

The visible entry shown in 'Add or Remove Programs' shall start with the name of the company.

# Bibliography

[1]     *NAMUR Worksheet Status Signals of Field Instruments*, NE107, Version 6.7.2005

[2]     *PROFIBUS Technology Glossary* http://www.profibus.org

[3]     '*Official Guidelines for User Interface for Developers and Designers*' Microsoft MSDN; http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/welcome.asp; 2005-02-03

[4]     Molich, R., and Nielsen, J. (1990). *Improving a human-computer dialogue*, Communications of the ACM 33, 3 (March), 338-348.

[5]     Nielsen, J. (1994b). *Heuristic evaluation*. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY.

[6]     *Microsoft Active Accessibility*, Version 2.0; http://msdn2.microsoft.com/en-us/library/ms697707.aspx; 2008-02-24

_____