Technology Overview

FDT® GROUP

Empowering the Intelligent Enterprise

Contents

1. FDT® Overview

1.1 Introduction

FDT® is an interface specification for open data exchange between field devices and automation systems that is standardized by the international standards IEC 62453, ISA 103, and GB/T 29618. In FDT, two terms are particularly important: DTM™ (Device Type Manager™, or "device driver") and FDT/FRAME™. Both are software components whose functions can only be performed together. Figure 1 shows the symbols used in the following texts and graphics.

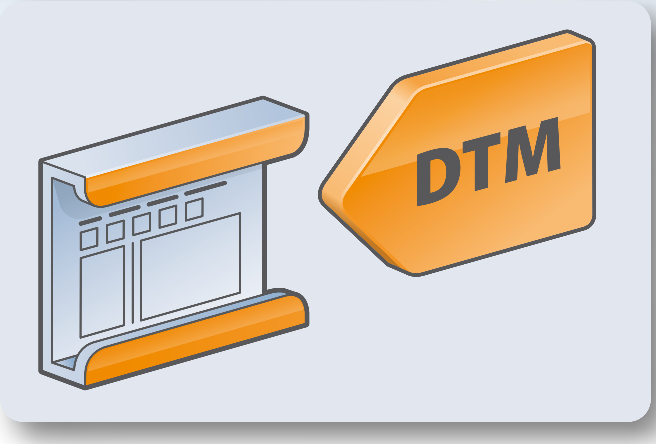


Figure 1: Symbols for FDT/FRAME (left) and FDT/DTM (right)

FDT provides a common platform for data exchange for all available device drivers (DTMs) produced under this standard. This allows complete and functional access across different network hierarchies for all functions for devices made available by the DTM. With this capability, every device can be configured, serviced, and maintained via one standardized user interface — independent of manufacturer, device type or communication protocol.

Information from a plant (especially communications for field devices) is needed throughout the entire lifecycle of a system or application. FDT provides support with versatile and extremely helpful functions as early as planning and project engineering, then during installation and commissioning, and finally during operation and service.

Requirements for device integration

Device integration has had great importance in automated industrial systems for years. Today's intelligent field devices feature — along with the process data — more and more information and functions that have to be made accessible to the automation system. Users therefore demand a standardized environment for central management, commissioning, configuration, and maintenance of all field devices. These must be independent of the device manufacturer as well as of the communication protocol used, and implement a seamless data exchange between the field devices and a higher level system, such as a control system or a plant asset management application. A standard for device integration should offer the user a free selection of all components installed in their system, independent of the manufacturer, and with it the option of being able to use the most suitable devices for their applications. Moreover, an open technology is needed that will protect investments of the plant operator in field device technology for the long term and that makes it possible for device manufacturers to create uniformity in device drivers running in different system environments.

FDT can fulfill both of these wide-ranging user and manufacturer expectations equally, because the FDT specification is the basis for the interfaces that were standardized in IEC 62453, ISA 103, and GB/T 29618.

A single DTM is required per device. As a result, the development costs for the device manufacturer will be lower. The intellectual property of the manufacturer is also effectively protected because the programmed content of a DTM in binary format restricts reverse engineering by third parties.

The FDT concept can in simplified terms be compared to a method used in office communications (see Figure 2). A good example here is a printer with a compatible device driver and an integrated graphic user interface, which appears in always the same fashion and with the same functions, for example in different office applications. With FDT, a device driver (DTM) is made available for the field device that allows access to the device via a graphical user interface.

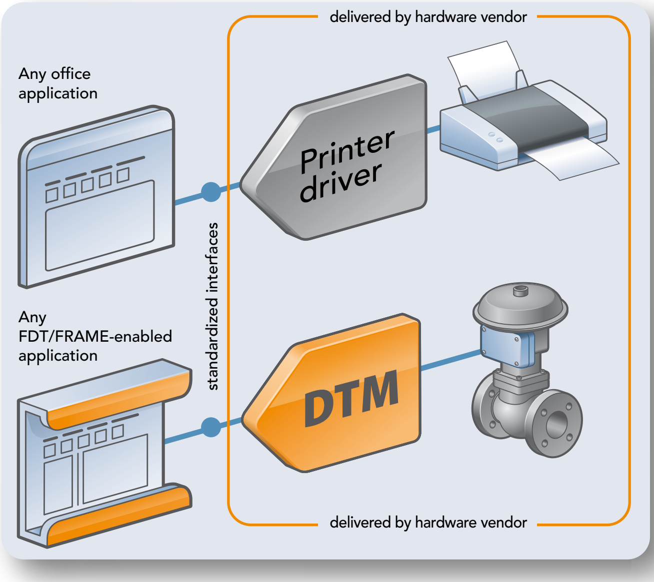


Figure 2: Printer driver and DTM in comparison

Through the FDT specification, general interoperability is ensured so that each DTM, regardless of manufacturer, can be run in every FDT/FRAME (see Figure 3).

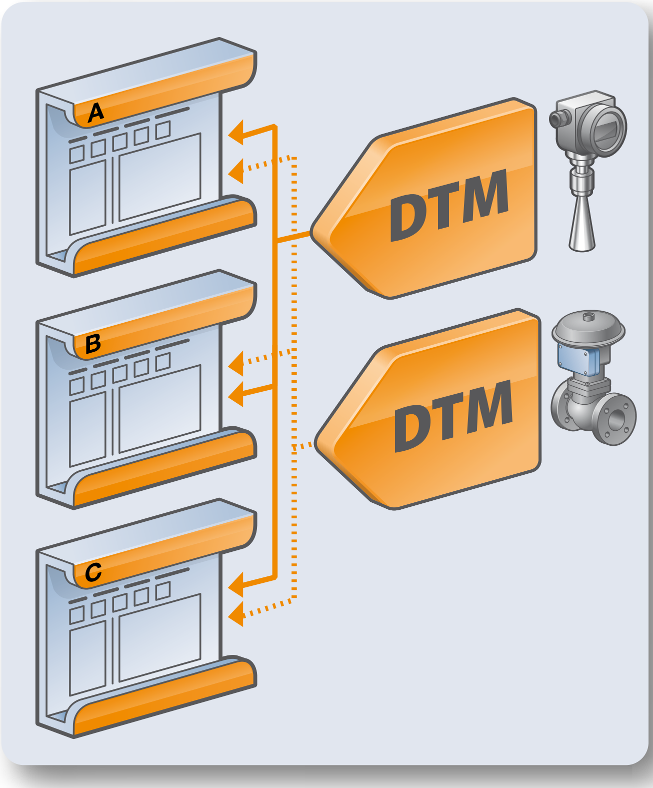


Figure 3: Interoperability between FDT/DTMs and FDT/FRAMEs

FDT Technology requires no specific adaptation of the field devices (for example, firmware or hardware) and can therefore be used universally for every device with a communications interface. The communication protocol supported by the device and the device properties are completely mapped by FDT Technology software on the PC. As a result, not only can new plants be equipped with FDT, but the technology also allows retroactive introduction of FDT in existing systems. When retrofitting an existing system, no modification or replacement of the installed device is required. The existing network of bus systems, communications facilities, and field devices may be used without change. The DTMs must simply be installed in the FDT/FRAME.

1.2 FDT Group

The FDT specification is administered, maintained and further developed by the FDT Group AISBL (Association Internationale Sans But Lucratif). This includes guidelines such as the DTM Style Guide and the FDT Lifecycle Policy. The FDT Group is an international non-profit association of leading global enterprises in process and manufacturing automation. The primary goal of the FDT Group is the global establishment of the FDT standard as an open, manufacturer-neutral interface description for integration of field devices in engineering, automation, and plant asset management systems. In this field, end users, manufacturers, universities and research facilities have come together for the common purpose of maintaining and expanding the technology. Development tools, support, training, test runs, and documentation are provided for this purpose.

The FDT Group ensures the quality and thus the interoperability of the products through a detailed certification test (see Chapter 5).

FDT Group continues to advance its standard to keep pace with the changing requirements of process, hybrid and factory automation, harnessing the Industrial Internet of Things (IIoT) and Industrie 4.0 for enterprise-wide network and asset integration for the "Connected World." The latest innovations empower the way industrial automation architectures connect and communicate sensor to cloud. These and other developments allow for faster execution for mega installations incorporating thousands of input/output (I/O) devices, as well as improved control system security on the wire.

The FDT standard has proven advantages to be effective for the new generation of smart operations, enabling improved configuration, calibration and diagnostics, and optimizing network interfacing and device configuration. Indeed, it will enable IIoT and Industrie 4.0 reality through an ecosystem of automation vendors promoting interoperability, security-on-the-wire, and mobility through tomorrow's new adaptive manufacturing assets.

Organizational structure

The FDT Group consists of the Board of Directors, an Executive Committee, and multiple department committees. The members of the FDT Group elect the Board of Directors, which in turn appoints the Executive Committee.

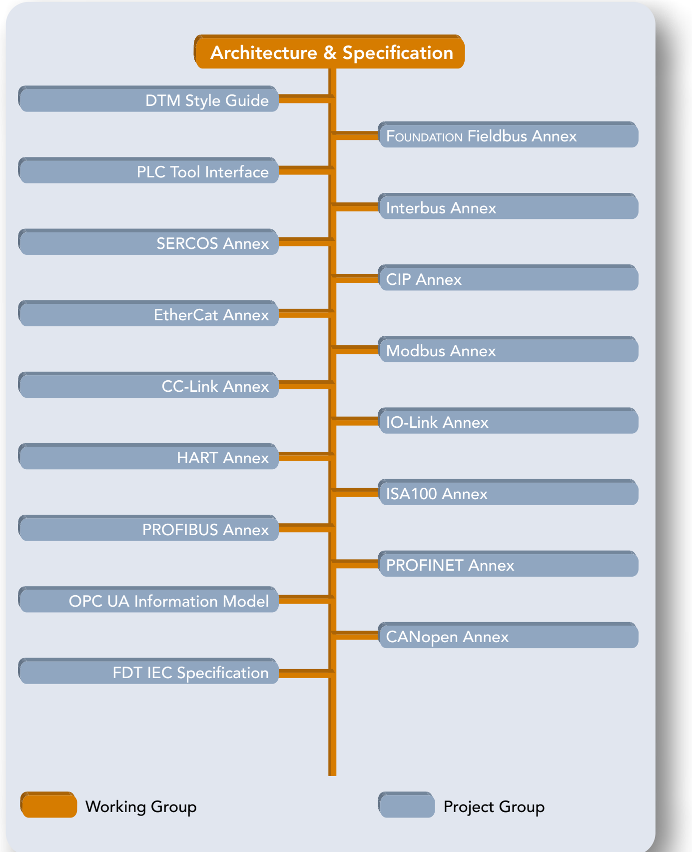


Figure 4: Project groups of WG Architecture & Specification

The other committees that carry out the work of the FDT Group are:

- Architecture and Specifications
- Associations and Standards
- Test and Certification
- Marketing

Working groups and project groups are a part of a committee. Figure 4 gives an overview of the working group "Architecture & Specification," the project group which deals with updating and expanding FDT Technology. When required, new theme-oriented working groups are approved by the Executive Committee, which work for example on protocol expansion (known as "annexes"). A project group is subordinate to the working group and exists only until the defined project objective is reached.

1.3 Historical development

The history of the FDT Group is one of advancement and innovation, as the organization has kept pace with the changing requirements of industrial measurement and control, and the diverse needs of automation suppliers, end users and other industry stakeholders.

In 1998, an initiative within the ZVEI (the German association for electro technical and electronic industry) started with the discussion on how to simplify the integration of field devices into distributed control systems (DCSs) with the availability of modern software component technologies. The PROFIBUS User Group adopted the specification results and later transferred the rights to the FDT Joint Interest Group, which was established shortly thereafter.

The FDT Group itself was started in early 2003 as an informal association by a number of leading automation firms: ABB, Endress+Hauser, Invensys (now Schneider Electric), Metso, and Siemens. With a growing number of members, it was decided to formally organize the group as a legally independent entity.

The FDT Group was officially founded in September 2005 as an International Not-For-Profit Association (AISBL) according to Belgian law. The FDT standard is internationally recognized as IEC 62453, as well as the North American standard ISA 103 and China GB/T 29618. All major leading manufacturing companies are now members of the group and contribute to a wide range of technology development and standardization initiatives.

From the beginning, the objective of FDT supporters was to standardize the communication and configuration interface between all field/communication devices and host systems. The FDT standard has continued to adapt to the realities of the process, hybrid and factory automation sectors, and today, it plays a vital role in the IIoT — optimizing industrial communication networks, automation systems, and device suppliers.

Today, hundreds of thousands of FDT/FRAMEs and millions of FDT/DTMs exist in the market. The FDT architecture focused on comprehensive integration ensures continuous innovation, optimizing reliable operations of plants, fabs, machinery, etc., all over the world.

1.4 FDT standard versions and supported protocols

FDT 1.2

The FDT 1.2 standard was released in 2001. Since that time, there has been one major enhancement — the downward compatible FDT 1.2.1 version, as well as several fieldbus protocol-specific adoptions that opened FDT Technology to fieldbus device manufacturers beyond HART and PROFIBUS. Foundation Fieldbus, Interbus, DeviceNet, and IO-Link were among them. Version 1.2.1 of the FDT specification uses COM and ActiveX as its basis technologies.

FDT 2.0

In 2012, the FDT Group announced the FDT 2.0 standard designed for the new era of automation, supporting the lifecycle of tomorrow's adaptive manufacturing assets for the "Connected World." Developed to leverage modern Microsoft.NET technology, FDT 2.0 supports an enhanced user interface with graphical representations of device parameters. It maintains proven FDT heritage, but includes numerous performance improvements. The technology provides backward compatibility with the existing installed base, eliminating "rip and replace" scenarios when utilizing different eras of field equipment allowing new and existing software to coexist.

FDT-supported networking protocols

FDT is built to support a comprehensive open architecture for the connected world of industrial automation networks and assets. The backbone architecture, standardized independent of network protocols, allows for a comprehensive integration model and a seamless integration mapping approach for connecting intelligent assets relaying device-specific diagnostic data enterprise-wide. All common networks for the process, hybrid and factory automation industries are supported by the FDT standard.

Figure 5: Logos of the protocols supported by FDT

2.1 Continuous device access across all levels

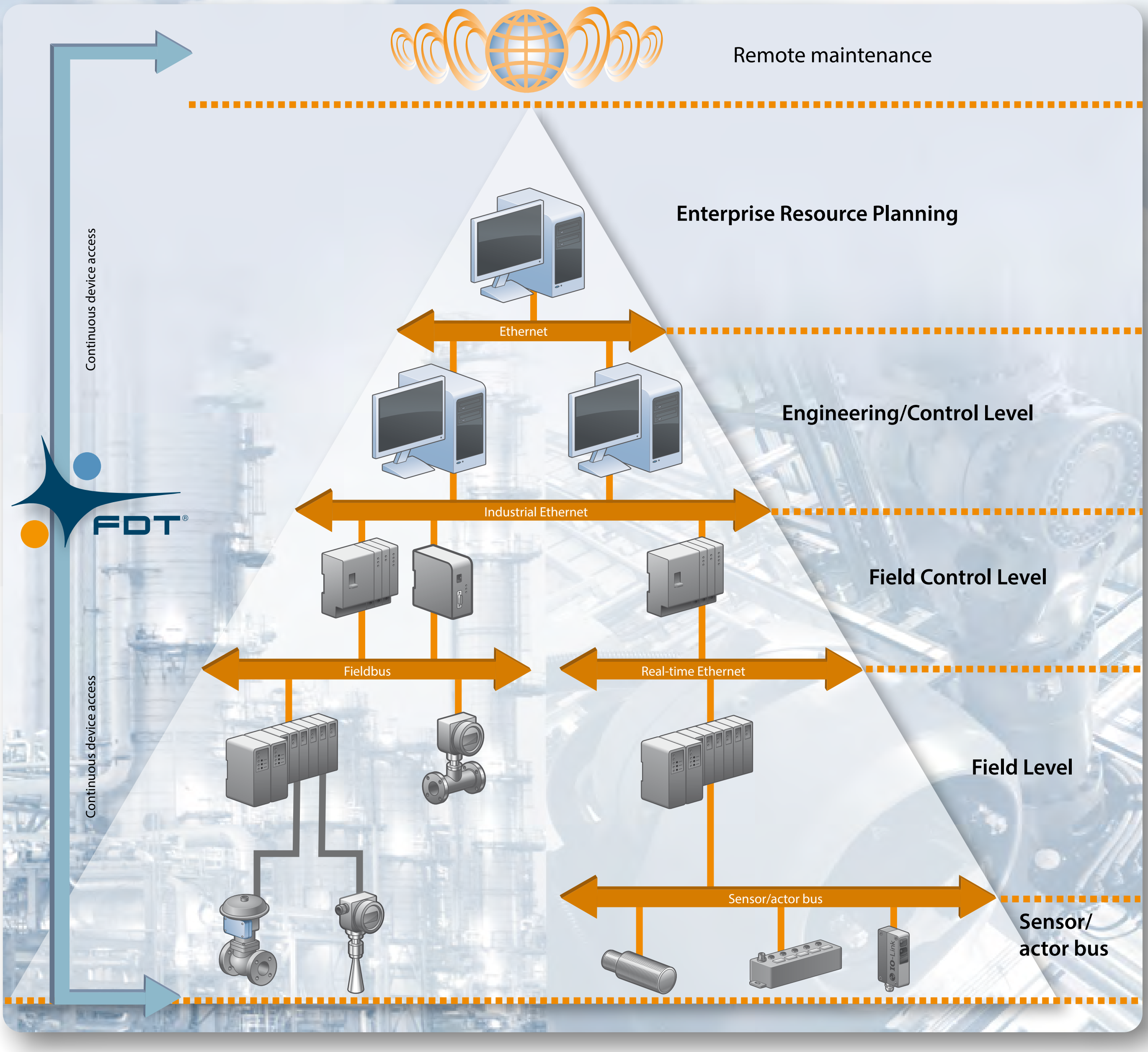


Figure 6: Automation pyramid

FDT Technology offers critical advantages, particularly in process and manufacturing automation plants and systems in which various forms of field devices and communication protocols are used on different levels of the automation pyramid (see Figure 6, right: manufacturing automation, left: process automation).

The functions of field devices from different manufacturers must often be adapted to the task at hand with special configuration software.

In addition, other valuable information is available over the lifecycle of system and field devices alongside the actual process data for system operation optimization. The greater the variety of the implemented device types, the more difficult and time-consuming their configuration will be. Users face particularly high costs and effort when device manufacturers provide different, non-standardized parameterization and configuration tools, and data must be exchanged between these tools and the engineering systems of the automation system. Here a data conversion is frequently necessary, which requires detailed, specialized knowledge. The consistency of data, documentation and configurations becomes very complex without FDT.

The urgent requirements of users and device manufacturers dictate that the tools used must be able to seamlessly integrate into the system tools of the automation system to make all system devices known to the central engineering system. In this regard, plant operators, plant engineers, maintenance engineers and device manufacturers find multilayered support through FDT.

Device and system integration can be configured with FDT across all levels of the automation pyramid and independently of the communications network and fieldbus using a single software tool. Protocol independence is achieved by the separation of the FDT basic specification and the protocol annexes. An annex describes the extensions of the FDT specification for the respective protocol.

For the plant operator of an automation plant, the working conditions are optimized through FDT, and the work sequences are made more efficient by:

- Providing a single, central workstation for engineering, commissioning, diagnostics and maintenance with direct access to all field devices
- Providing continuous access to all devices, independent from the given network structures and communication protocols, from a central location or from a decentralized one
- Exchange of common data between system tools (engineering or plant asset management systems) and device-specific user interfaces.
- Central data management and security as well as management of multi-workstation systems.

Figure 7: Use of FDT along the plant's lifecycle

2.2 FDT — More than just device access

FDT supports much more than just configuration and parameterization of devices. For each phase of the lifecycle of an automation system, such as planning or installation, specific functions are needed that are made available by FDT Technology (see Figure 7).

The following examples show some sophisticated implementation scenarios for FDT.

Offline engineering

In contrast to online engineering, which is feasible only in connection with the existing infrastructure and field devices, FDT makes offline engineering possible, even in the engineering phase (in other words, before the devices are present). For this purpose, the configuration data of the devices is entered into the associated FDT/DTM and saved in the project file of the FDT/FRAME. After successful installation of the devices, the data can be downloaded to them. FDT consequentially noticeably reduces the commissioning time of a system.

Network scan

FDT offers functions for automatic detection of the field devices connected to the bus. Through a communication DTM, the lower-level network topology, including the connected devices, can be identified. These functions can be used for automatic generation or verification of the system topology. Additionally, during a network scan, the fieldbus addresses of identified devices can be set.

Topology import/export

FDT defines a method of exchange of topology information between FDT/FRAMEs. Application-specific data from the respective DTMs is also contained in the export information. This function can for example be used to archive historical data records of DTM instances.

Audit trail (process and device history)

All engineer and operator intervention can be traced in a log file where it can be documented which user made what changes when and why. Through this log file, historical data can automatically be traced and documented; the audit trail becomes a helpful tool for validation processes.

Persistent data storage

The FDT/FRAME makes a uniform project database available with persistent data storage of all devices that are integrated into an automation system with DTMs. The persistent nature of DTMs ensures that when a previously configured device is accessed, its data is represented with the previously configured data intact. The project data can then be centrally backed up and restored. The database allows data storage and data backup for all devices of a system in a single project file.

User management

FDT specifies a uniform user management model with specific access rights for defined users or user groups: observers, operators, and engineers. Thus, it is ensured that specific sensitive functions of the FDT/DTM or the FDT/FRAME are only accessible to authorized users.

System documentation

The parameters and "type plate information" of the installed devices are needed for documentation of the system. For this purpose, FDT specifies a uniform interface over which this data can be electronically called up, and through which the same design can be output in easy-to-read form.

Diagnostics and online values

DTM allows online access during the runtime of the system and displays diagnostics, status monitoring, and measured values of the devices or the process.

2.3 Benefits for plant operators

The user benefit from FDT is multi-layered and is additionally reflected in the above-named scenarios — corresponding to the possible applications — across the full lifecycle of a device and/or system.

- **Freedom of device selection**

  Users can freely select various manufacturers' devices available on the market and integrate the best-fitting devices for their applications. The selection can be oriented towards the performance features of the device and do not need to follow the specifications of a system manufacturer or a communication protocol. The same applies for the FDT/FRAME.

- **Higher efficiency**

  The entirety of the field device data can be called up and all common tasks can be supported across the entire lifecycle of a device, from offline engineering before the commissioning to diagnostics and maintenance information for predictive maintenance. The advantages are shortened installation time and higher plant availability.

- **Assurance in plant operation**

  The user interface of all DTMs follows the guidelines of the DTM Style Guide in its configuration, for example, in the separation of application areas. This provides greater assurance for device operation and reduces training effort. A variety of operating tools can be replaced by a single FDT/FRAME.

- **Openness and future-readiness**

  FDT imposes no limitations on an innovative device technology as long as the DTM interface remains compatible. Investments are thus protected long-term. New communication protocols are regularly added to the FDT specification and backward compatibility of new interface descriptions is a requirement for all FDT working groups.

2.4 Benefits for device manufacturers and system providers

FDT Technology also offers significant benefits for device manufacturers, system providers and end users:

- **Reduced development effort with greater capabilities**

  Thanks to FDT Technology, the device manufacturer only has to develop a single device driver (DTM) that can be run in every FDT/FRAME. High costs for multiple developments due to varying system environments are omitted.

  The FDT/DTM can be designed to have any desired level of performance (for example, to support particularly intuitive user interfaces or complex algorithms). There are no limitations in regard to the DTM-supported performance characteristics. All functions that can be realized with a modern programming language can be implemented.

- **Investment and expertise protection**

  The device manufacturer's investments are protected, because a change to the system environment will not require the components to be modified. Device suppliers can instead concentrate on developing special performance characteristics for their devices, with which they can stand out from the competition by offering their customers critical benefits.

  The expertise of the manufacturer remains reliably protected, as the DTM is shipped in binary format.

- **High efficiency and flexibility**

  Existing device description files, such as EDD or IODD, can be converted quickly and efficiently into a DTM which can be used in all FDT/FRAMEs. Development tools for this are available on the market. In addition, interpreting DTMs exist to perform the same conversion in the FDT/FRAME.

  System manufacturers can integrate uniform device management through FDT Technology for different communication protocols in the individual system tools.

  Through the FDT-specified functions and interfaces, all data points supported by the device are transmitted to the fieldbus master. The DTM user can make changes and apply them directly to the engineering application.

3. Implementation of FDT

FDT Technology comprises two software components, the FDT/FRAME and the FDT/DTM, which interact via the interfaces standardized in the IEC 62453, ISA 103, and GB/T 29618.

3.1 FDT/FRAME

An FDT/FRAME, whether it's a device configuration tool, system engineering tool, operator console, or asset management tool, provides access to all devices, gateway and communication components with single-point access to operational assets. With the aid of this component, the information of processes in the field is accessible in real-time and available across the lifecycle of a plant or application.

The FDT/FRAME provides a common runtime environment and renders a graphical user interface to DTMs, and is the unifying tool embedded in system/host applications like asset management tools, Programmable Logic Controllers (PLCs) or Distributed Control Systems (DCSs).

The FDT/FRAME is designed to administer entire device instances and save their data without the need to possess device-specific knowledge. This guarantees enterprise-wide, uniform lifecycle management, and supports multi-workstation and single workstation environments.

3.2 FDT/DTM

An FDT/DTM is a software component with content that represents a certain device or a communication component with its/their properties and functions, including the user interface. The DTMs organize the handling from an FDT/FRAME with "their" devices and communication components found in the system. A user can thus call up the functions of the connected devices through the associated DTMs. A DTM can, with respect to its function, also be seen as a "Proxy" that provides control over a device or a communication component in the plant over the FDT/FRAME. In this regard, it is also referred to as an instance in the FDT/FRAME. The DTM is not a stand-alone program; rather, it always requires an FDT/FRAME to run.

FDT/DTMs make it easier for users to work with field devices, as no special knowledge of the bus systems being used and their communication protocols between operator workstation and the field is needed. All devices present themselves in the same manner since essential parts of the user interface of the DTM are laid out in the FDT Style Guide (see Chapter 3.3). This is true for device-specific as well as for generic DTMs, which will be more fully described in the next section. Through this, an intuitive and simple operation and configuration of devices is possible. Corresponding to their functional assignment, a distinction is made between a device DTM and communication DTM (see Figure 8).

Device DTM

Device DTMs (see Figure 8) have a device-specific structure and contain data, functions and logic elements of the device. Depending on the degree of implementation, a DTM contains features ranging from a simple graphic user interface for setting the device parameters, to integrated applications for diagnostics and maintenance purposes, such as the logic for device calibration. Device DTMs are usually developed by the device manufacturer and provided as a part of the delivery package. In contrast to the design freedom for the content of a DTM, its interface to the FDT/FRAME is defined in binding form by the FDT specification.

Device DTMs are divided into three groups, depending on their use:

- **Manufacturer-specific DTMs** — These DTMs offer the greatest functionality and support. A single device DTM can support one or a family of common types of devices such as pressure or temperature transmitters. These DTMs can be expanded with additional features such as extended configuration and diagnostic capabilities, networking analysis or curve display.

- **Interpreter DTMs** — These DTMs are not uniquely programmed for a specific device. Instead, they interpret these other types of device representations, for example: Device Descriptions (DDs), Electronic Device Descriptions (EDDs), Field Device Integration (FDI) Device Packages, and IODD to allow configuration and parameter access within an FDT/FRAME-enabled system.

- **Generic (Universal) DTMs** — A Generic DTM has the ability to universally represent all devices with the compliant parameters of a specific protocol, creating simplicity in the architecture employed by a single DTM within the FDT/FRAME system. As an example: With a generic HART DTM, these are the parameters that can be reached under the term "Universal and Common Practice Commands."

DTM for communication components

Comparable to the representation of field devices through specific device DTMs, the network infrastructure components participating in the communication such as communication adapters, gateways or bus couplers are also represented by corresponding DTMs. The communication channel is a shared feature of this DTM; the DTM makes the communication channel available to the assigned device DTM and maps the specific services of the network protocol into corresponding software interfaces.

Depending on the execution of the associated hardware components and operating location in the network, two DTM categories are possible:

- **Communication DTMs (CommDTMs)** — A CommDTM is the first DTM to be activated upon communication setup in an FDT/FRAME system. It standardizes the communication channel to the corresponding communication operations of the mapped network protocol. The communication DTM consequently acts as the standardized device driver for one or multiple protocols.
- **Gateway DTMs** — The Gateway DTM allows communication to transition between the communication paths of different protocols in the FDT architecture. They act as a link between the communication DTM and device DTM, allowing the cascade of information being communicated to move bi-laterally, including transitions between different protocols, such as Ethernet-based systems and PROFIBUS or IO-Link. They are therefore the basis for building vertical communications across system boundaries (see Chapter 4.5).

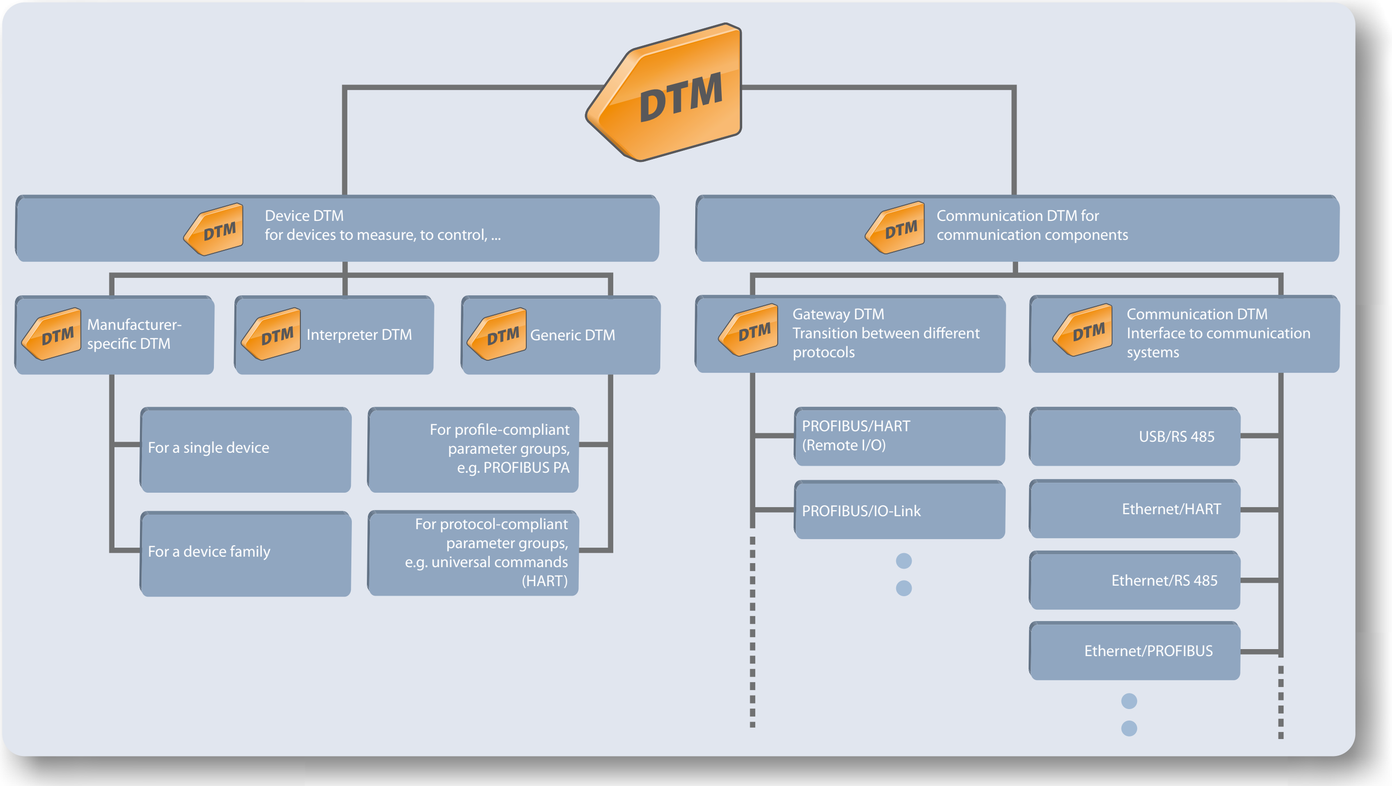


Figure 8: DTM categories

3.3 Designing DTM user interfaces (Style Guide)

Uniform look & feel

The graphic user interface of the DTMs is the everyday tool for users of FDT Technology that makes a substantial contribution to efficient work with its easy-to-understand and uniform structure across all manufacturers and device types. Therefore, the FDT Group has developed a DTM Style Guide that outlines the rules for the structure (look & feel) of the user interface. The rules are also set forth in a checklist that can serve as a guideline for DTM developers. Style Guide compliance is indicated on the certification test results.

The objective of the FDT Group is to ensure that all basic functions of a DTM run the same and have a uniform appearance while not imposing limitations on task and process-oriented views. For this purpose, the Style Guide prescribes the division of the user interface into general and task-related areas, provides a library of icons and their meaning and provides a glossary of terms and standard messages in at least eight languages. Thus, users encounter the listed components in all DTMs with the same appearance and the same meaning.

Supplementary requirements of the Style Guide for the DTM developers are the ability to reach all elements by keyboard as well as a Microsoft Windows-type behavior.

Division of the user interface

The DTM Style Guide defines a scalable DTM user interface with defined areas (see Figure 9):

- **Identification area**

  This area helps the user uniquely identify the field device. In addition to the device and company logo, it may include information about the device designation, instrument number, and the version information. DTM-specific information is optionally available.

- **Toolbar (optional)**

  This bar provides buttons that can be used to gain quick access to frequently needed functions such as Print, Copy, Paste, Diagnostics, etc.

- **Navigation area (optional)**
  This area includes the various instrument functions, arranged via an easy-to-understand tree structure. In this structure, users can easily navigate and reach the function they need such as configuration, diagnostics or maintenance.

- **Application area**
  This area is used to display DTM functions that the manufacturer requires for specific, application-oriented tasks. All functions can be processed in graphic form (including 3D) and enable the display of complex functions, such as measured value curves, tables, auxiliary functions, etc.

- **Action area (deployed by the FDT/FRAME)**
  This area has buttons for running certain functions, divided according to application-oriented and standard functions such as OK, Close, Back, etc.

- **Status bar (deployed by the FDT/FRAME)**
  This bar is used to display general DTM statuses such as online access, user rights, parameters changed, etc.

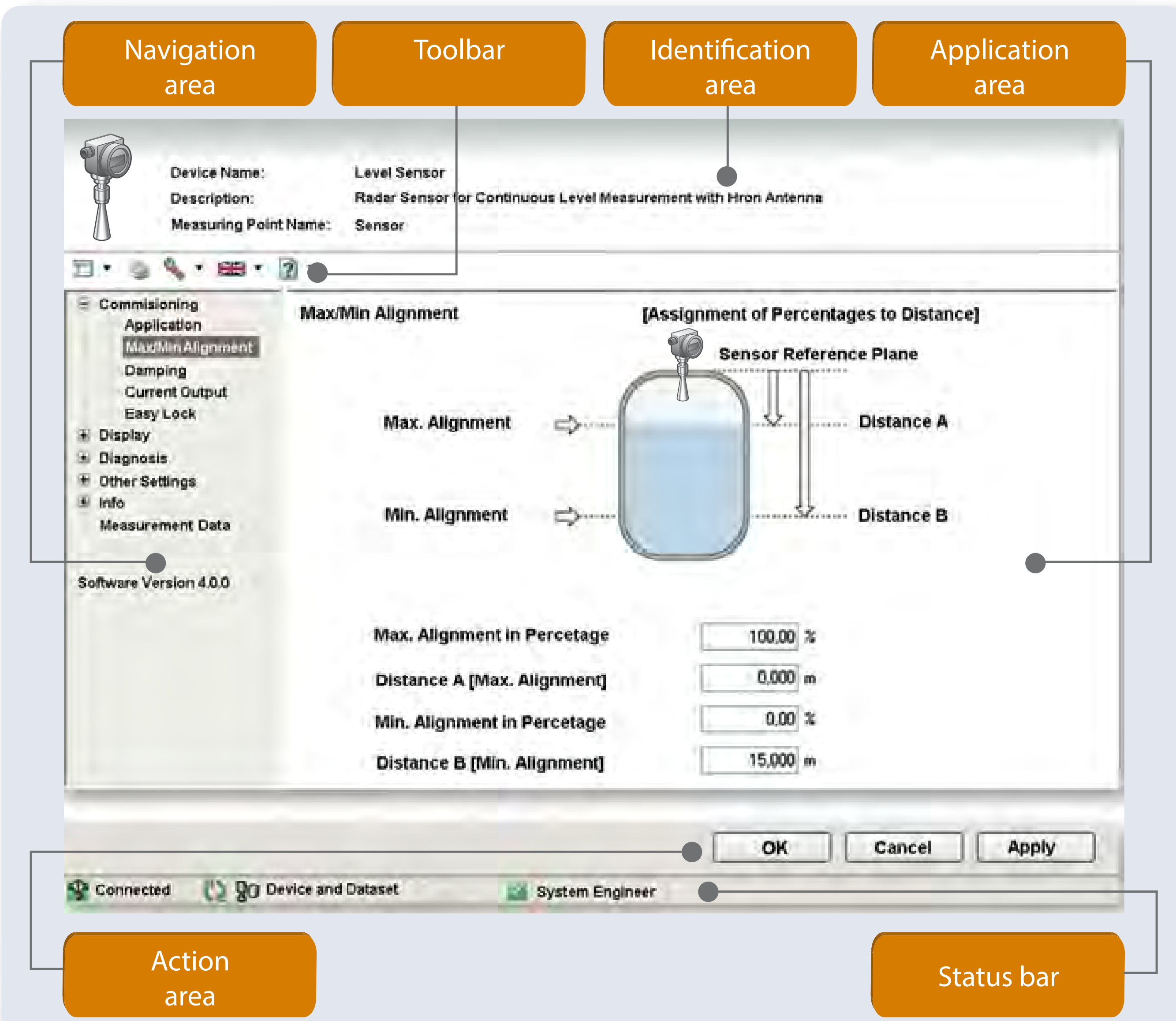


Figure 9: User interface of a DTM with Style Guide-compliant structure

3.4 DTM applications

As shown in Figure 7, DTMs are used over the entire lifecycle of a plant. Manufacturer-specific device functions such as status monitoring and status messages, interactive functions and graphic objects, trend curves, echo curves, time curves or device-specific online help features can be integrated into the DTM. Using the DTMs, the device manufacturer has the ability to provide users with a wide variety of types of support, from commissioning tools or diagnostic options to maintenance and repair instructions.

Plant design and project engineering

For design and project engineering tasks, the FDT/FRAME has an easy-to-understand and clearly structured device catalogue. The user may use filter functions to adapt the view of the device catalogue, making the designer's work easier by providing the best possible view of the plant with all its devices. For example, a project view can show the structure of the plant from the standpoint of communication between PC and field devices (routing of the communication paths). The plant view can take into account the technical aspect of a plant or plant unit. In addition, colored symbols can be used to indicate the current state of a plant, making maintenance and troubleshooting faster.

Commissioning

Device configuration is a key task in the engineering phase of each project, small or large. The configuration function configures the device parameters that can be defined even without a connected device (offline). The parameter set is created or changed and stored in the project. When a field device is connected during commissioning of the project, the parameter set can be loaded to the device. In this way, substantial time savings during commissioning can be achieved. The online configuration function, on the other hand, is used to change parameters when the field device is connected and to verify its correct functioning.

Plant operation and maintenance

For plant operation and maintenance, time-optimized work is the rule. FDT/FRAMEs support simultaneous editing of multiple devices in a single work step. After the relevant devices are selected, the desired function (e.g., reading out the device status) is started, which is then carried out automatically for all selected devices. With the same objective of time optimization, for example, the often-lengthy process of reading and writing parameters in field devices can be carried out automatically and for multiple devices or a certain plant unit in a single work step. Plant operators and maintenance personnel often have the same need for actual status information of the field devices in the plant. Using the FDT/FRAME, a predefined quantity of field devices can be polled for their status information cyclically or on a one-time basis. In doing so, filter functions enhance ease of understanding and make display and evaluation easier. Instead of a complete log, for example, the user can view and output only a certain field device or only a certain status value.

Following the market requirements of the process industry, specifically the diagnostics classification of NAMUR Recommendation 107 is taken into account here as an option, including use of the corresponding symbols (see Figure 10).

Device status	Meaning according to NE 107 (NAMUR)	Symbol
Failure	Output signal invalid due to malfunction in the field device or its peripherals.	
Function check	Output signal temporarily invalid (e.g. frozen) due to ongoing work on the device.	
Out of specification	Deviations from the permissible ambient or process conditions determined by the device itself through self-monitoring or faults in the device itself indicate that the measuring uncertainty of sensors or deviations from the set value in actuators is probably greater than expected under operating conditions.	?
Maintenance request	Although the output signal is valid, the wear reserve is nearly exhausted or a function will soon be restricted due to operational conditions e.g. aging of a pH-electrode.	

Figure 10: NAMUR classification of the device status classes (NE 107)

This allows device-specific events such as maintenance required or failure, to be displayed to the user in easy-to-understand form using the symbols of NE107.

The error monitor in the FDT/FRAME helps to ensure proper functioning. As soon as a DTM identifies an error in processing a function, an error message is output and entered into a list that contains all error messages since the start of the FDT/FRAME or since the last time the error list was acknowledged. Finally, the debug monitor serves to analyze and document error states, which classifies the results into four message classes and with specification of the source and time stamp.

FDT defines the following types of messages in this regard:

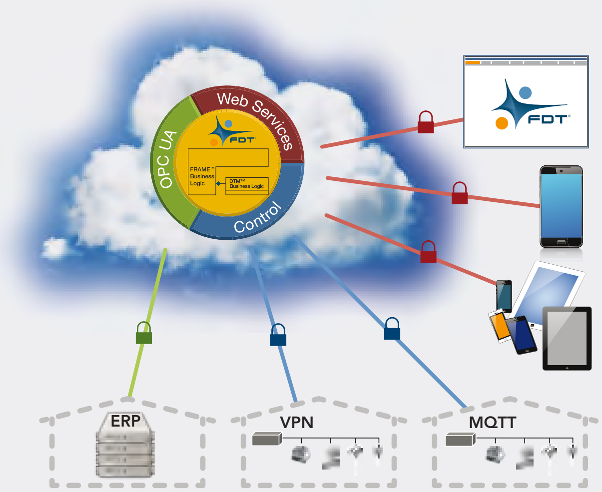
- Critical
- Error
- Warning
- Information
- Verbose

The messages can be sent by the DTM in any situation and enable the DTM developer to analyze errors quickly and debug efficiently.

FDT IIoT Server (FITS™) — Simplifying the Automation Ecosystem Exchange

Today, plants and factories employing FDT-enabled systems are already benefitting from open access to the Industrial Internet. To advance its support of the IIoT and Industrie 4.0, FDT Group has developed a solution known as "FITS™" (FDT IIoT Server). It enables mobility, cloud, and fog enterprise applications, as well as sensor-to-cloud and enterprise-wide connectivity employing FRAME™ and DTM™ business logic at the heart of its client-server architecture.

Through FITS, FDT Group is committed to making the IIoT a reality via a broad ecosystem that spans the process, hybrid and factory automation markets, and involves controls and instrumentation suppliers, end users, standards organizations, etc. — all aimed at promoting interoperability, security and mobility through new, adaptive manufacturing assets.



FITS represents the next generation of technology intended to protect legacy investments in the FDT standard through advanced business logic, well-defined interfaces and common components, while also providing the foundation for a modern, integrated automation architecture.

The FDT architecture (based off of the current FDT 2.0 standard) is being enhanced to include operating system (OS) agnostic support for standard browsers, fit-for-purpose apps, and general Web Services utilizing the latest generation of FRAMEs and DTMs. Current FDT-enabled systems support an IIoT-connected enterprise by creating a single system infrastructure that standardizes the connection of disparate automation assets.

Learn more about FITS by visiting www.fdtgroup.org.

4. FDT Specification & Architecture Evolves

4.1 FDT today — Integrating tomorrow's technology

The FDT Group continues to advance its free and open standard to keep pace with the changing requirements of process, hybrid and factory automation for the new era of automation. The following criteria were market-driven requirements to evolve the standard from FDT 1.x to the current active version of the standard FDT 2.0:

- Maintain proven FDT heritage which frees automation end users from the constraints imposed by field communication protocols, control platforms, instrument suppliers, and the manufacturing environment
- Comprehensive interoperability for heterogeneous control architectures, allowing best-in-class network and asset integration, as well as the cascade of information from sensor to the enterprise
- A robust integration platform solution allowing for backwards compatibility bridging legacy and the new era of automation, making it easier to access information in plants and facilities with multi-generational assets that use one or multiple communication networks
- Allowing for a scalable and secure integration approach expanding the existing degree of openness to adapt any field communication protocol that may be needed along with the incorporation of new standards for enterprise connectivity, mobility applications and others as the future demands
- Provide investment protection, delivering a comprehensive IIoT architecture for the industry through the ecosystem of automation vendors and enabling seamless interoperability, security and mobility through tomorrow's new adaptive lifecycle manufacturing assets for the process, hybrid and discrete user community

4.2 Evolution from COM to .NET

FDT 1.x and COM

FDT has made consistent use of industry standards and selected the Microsoft® COM (Component Object Model) platform for software components. COM features a tried-and-tested client-server architecture and manages the integration of the software components into the FDT/FRAME. COM enables communication across processes and dynamic object creation. A COM component makes its function accessible via interfaces, which was used to define interfaces in FDT/DTMs and FDT/FRAMEs. Graphic user interfaces (GUIs) are set up using ActiveX® technology, an extension of COM technology that defines how GUIs are integrated into an application. In FDT, the ActiveX control element is provided by the FDT/FRAME and is connected to the DTM for data exchange. Thus, it integrates seamlessly into the user interface of the FDT/FRAME and can still provide the entire range of DTM functions.

For data exchange between objects, for example, between the FDT/FRAME and FDT/DTM, FDT 1.x uses XML (Extensible Markup Language), a standard for creating data documents in a hierarchical structure.

FDT 2.x and .NET

Over time, the FDT Group has developed a concept to further the development of its standards with consideration to future technologies in software development. As a result, a future-proof architecture concept for FDT 2.0 came into being based on .NET.

The .NET technologies are based completely on open standards and specifications, and offer all the properties needed by up-to-date software right up to web and client/server based systems. And with Windows Presentation Foundation (WPF), Microsoft has created a technology to create sophisticated user interfaces in 2D or 3D. .NET technology is based on runtime: the program code is interpreted at runtime and converted to machine code. The .NET execution environment, Common Language Runtime (CLR), is a concrete implementation of international standardized CLI (Common Language Infrastructure). This produces a certain measure of platform-independence. A crucial point is the backward compatibility of .NET with older Microsoft technologies. For example, .NET can use "older" software like COM/ActiveX components and it is even possible to develop such components with .NET. The compatibility also functions in both directions. These advantages have led to .NET finding applications in many industrial areas. .NET has also already found uses in the development of later versions of FDT 1.x software. Consequently, the further development of COM/ActiveX® up to .NET is a natural evolutionary step for FDT as well. An unlimited support of 64 Bit operating system also applies here.

4.3 Implementation of FDT 2.0

Same basic concept

The proven values of the basic FDT concepts are completely retained in FDT 2.0 and simply transferred to the new technological platform (.NET). The device manufacturer further supplies the DTM with device-specific functions together with the device, and provides a user interface as a driver for integration in an FDT/FRAME. The DTM user interface makes it possible for the user to change device parameters or to execute other functions.

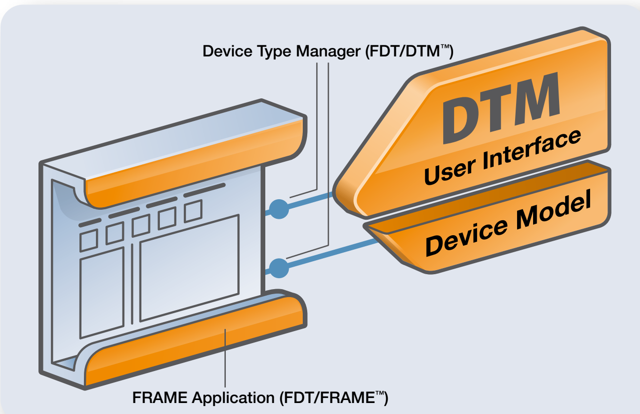


Figure 11: DTM corresponding to FDT 2.0

Suitability for distributed systems

In contrast to version 1.x, the execution of DTMs in "DTM device logic" and "DTM user interface" is separated in FDT 2.0 (see Figure 11). Similarly, the interaction of the DTM and its components is no longer direct between the DTMs using private interfaces as previously, but rather it is done exclusively over the interfaces as specified by FDT 2.0 and provided by the FDT/FRAME. Now distributed systems can be very simply implemented with execution of the DTM user interfaces on spatially separate client computers in conjunction with DTM device logic on a single centralized server. The DTM device logic is then implemented by a ".NET class" and the DTM user interface as .NET WPF or WinForms Control. The interaction with the application is carried out over .NET interfaces. Great care has been taken to ensure that these interfaces are as simple to configure as possible.

Automatic installation

In FDT 1.x, the DTMs are manually installed, whereby the correct DTM for the device type/version in question must be searched for and then installed on every computer. These limitations do not apply with FDT 2.0, as the DTM setup always provides the necessary information and functions for automatic installation.

Greater performance

In FDT 2.0, care was taken to ensure improved performance of FDT applications through numerous design measures. The call-up times and the allocation of RAM on the PC were specially optimized.

Catalogue update

After installation or uninstallation of DTMs, the changes are more quickly available in the DTM catalogue of the FDT 2.0 FRAME.

Partial loading and saving of DTM-specific data

Parameters are divided into data blocks and can be dynamically loaded during runtime. One instance does not have to immediately load the entire data record of a DTM, rather it is initially started with the minimum required data and parameters of the DTM user interface called up. Only if the user calls up further user interfaces are the additional data records loaded. This advantage is especially apparent in FDT 2.0 FRAME projects with many DTM instances. First, only the DTM type information is loaded then the loading of the DTM data instance occurs only with corresponding access.

Calling up over the FDT interface

With FDT 2.0, no XML documents will be exchanged over the interfaces; the exchange of data and information occurs through .NET objects. The relatively frequent creation and parsing as well as filling of the internal data structures with information from the XML file are thus eliminated.

Interface to the PLC engineering system

An interface for the PLC programming environment, called the PLC tool interface, is also among the new features of FDT 2.0. The process map of a device can thus be configured over the device DTM and checked over the PLC engineering system before the changes are activated in the PLC. Further details are described in the chapter "PLC Tool Interface."

Ensured interoperability

Interoperability among various FDT components has been improved by simplified interfaces and also through detailed specification. Many valuable experiences of users of the previous versions of FDT influenced this design feature.

Interoperability is in principle dependent on specification compliant implementations by the specific products. In this context, by developing standard components ("Common Components"), the FDT Group has taken a critical step forward. With direct use of Common Components, new development of FDT/DTM and FDT/FRAME applications can now be done more quickly and cheaply, and complex compatibility and interoperability tests can largely be eliminated.

Security

The topic of security in the sense of access and tamper protection is becoming more significant in the software industry. With industrial software, the risk to system stability is mostly in the foreground, and "criminal efforts," such as that faced by web applications, is confronted less often. This risk is generally confronted in the industrial automation industry through the IT infrastructure and restrictive IT guidelines.

This approach to system stabilization applies also in FDT 2.0 in two ways:

The FDT/DTM manufacturer can become certified by a member of the "Windows Root Certificate Program" and then digitally sign their DTM delivery with regard to origin and authenticity using Microsoft Authenticode® digital technology. The FDT/FRAME can verify this signature during DTM registration and thus confirm the quality and reliability of the DTM.

Furthermore, the FDT Group has set up a certification process for DTMs regarding their FDT conformity. Certified DTMs contain a digitally signed voucher from the FDT Group. The private cryptographic key for creating these digital signatures is known exclusively by the FDT Group. The FDT/FRAME can establish the validity of the certificate of conformity, as it can check the signature by means of the corresponding public key of the signing entity (FDT Group). Both methods together grant a high degree of security in the use of DTMs.

Support for the plant lifecycle

Aspects of the plant's lifecycle on device integration in systems are an integral part of FDT Technology. In FDT 2.0, this topic is dealt with intensively by means of various definitions and guidelines. An example is the implementation of a device exchange with the accompanying update of the DTM. This is a function that is handled as a special case in FDT 2.0.

The required backward compatibility of FDT 2.0 to FDT 1.x is ensured by FDT 2.0 and by the manufacturers. The interoperability between FDT 1.x DTMs and FDT 2.0 DTMs is a part of the FDT 2.0 specification and is thus safeguarded as technically viable. Consequently, every FDT/FRAME developed according to specification FDT 2.0 ensures this backward compatibility. This means that FDT 2.0 DTMs and FDT 1.x DTMs are supported (see Figure 12).

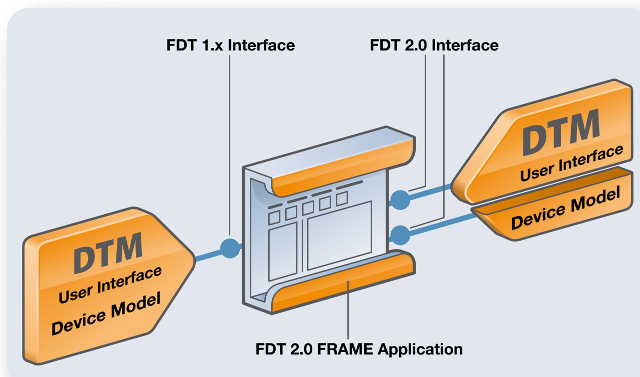


Figure 12: Compatibility of FDT 2.0 and FDT 1.x

Expanding enterprise connectivity

The FDT/OPC UA companion specification, a plug-in for the FDT 2.0 standard, provides connectivity to enable data communication throughout the enterprise. This approach exemplifies the device-to-cloud computing strategy that supports configuration, diagnostics, asset health, communication, historical data access, and alarming and event services for existing and upcoming devices that have FDT support.

The FDT standard incorporates a plant hierarchy based on physical network topology coupled with a logical topology. The network protocols in the industry allow an FDT/FRAME application system to talk with any device. This includes the ability to transparently tunnel through disparate networks to access the end device. The FRAME also receives operational lifecycle access to commissioning, diagnostic, prognostic, and other high-level data. To make these data sources available to the OPC UA architecture, the data has been mapped to the OPC UA data model, and the FDT/FRAME enabled system is configured as an OPC UA server.

The client can request a secure connection with the FDT/OPC server and access topology, health, and other data. Any number of clients may gain access, limited only by the server's capacity and underlying network bandwidths.

Logical and physical topology

FDT distinguishes between two different topological views, the logical and the physical. Figure 13 displays the real machine on the left. The physical topology view in an FDT/FRAME (center) represents the corresponding physical connections, such as cable lines or wireless connections, between the devices and the corresponding FDT/DTMs, and as such, describes the hardware installation of the machine. The logical topology view (right) displays the corresponding familiar hierarchy of the DTMs in an FDT/FRAME project and describes the communication relationships among them. The physical topology can be used to provide any kind of network structures. Although optional, the FDT/FRAME is responsible for managing the physical topology. On the other hand, an FDT/DTM must always make all the information about its hardware interfaces available so that an FDT/FRAME can establish whether a certain physical connection is possible or not. The physical topology has no dependence to the logical topology and is managed separately by the FDT/FRAME.

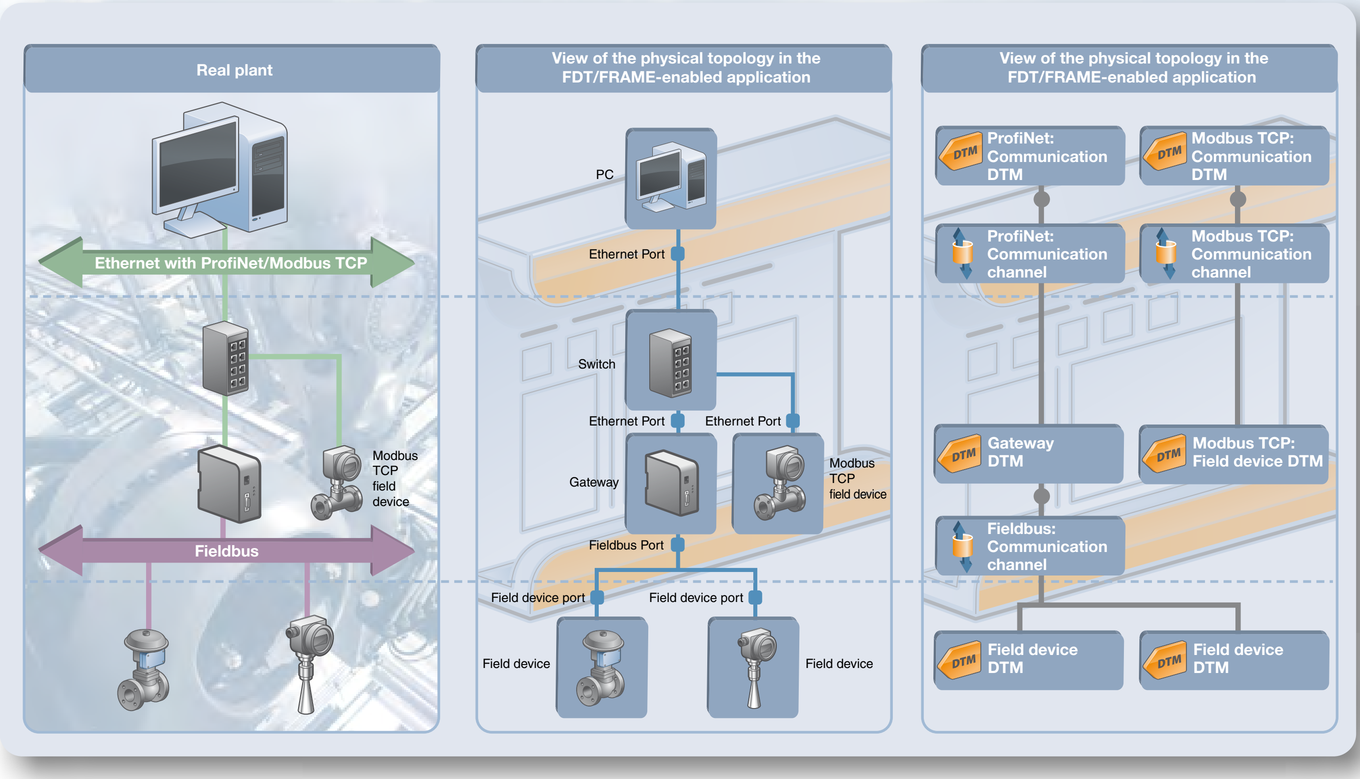


Figure 13: Real plant with corresponding physical and logical topology

Passive devices

From the FDT point of view, passive devices (non-communicating devices) such as power supplies or terminators do not participate directly in the online communication of parameters and process data. However, via specific properties, they can influence the modeling of physical network topology (e.g., line lengths, transmission rates, maximum number of nodes) and the engineering process. Thus, a relationship with FDT emerges. Properties of such passive devices or their DTMs are, for example:

- Connection does not take place over a fieldbus protocol
- Online configuration via FDT mechanisms is not necessary and/or not possible
- Process data is not made available

Certain properties of passive devices are made public via protocol-specific extensions. For instance, additions to the protocol for a backplane bus, which is used for communication with individual connected modules, should be defined for a power supply module with a modular IO device. In an FDT application, this is usually a manufacturer-specific protocol.

Even an "active" device (communicating device) can make engineering-relevant properties available in this way. Over the bus, both the power supply modules ("passive" devices) provide their maximum output load and field devices ("active" or communicating devices) provide their maximum current consumption for a bus system with a power supply for field devices. In this way, an engineering or diagnostic tool can check both parameters against each other.

Static functions

Current trends of plant management (e.g. condition monitoring and plant asset management) require the option of retrieving the necessary information with little effort from all of the system's devices and to make them available to plant management. The static functions mechanism in the FDT 2.0 specification is used to meet this requirement. It is the basis for online monitoring of devices.

Static functions are provided through a DTM, but can be carried out independently of its runtime. In this way, it is possible to retrieve individual information necessary for evaluating the device's status independent of the DTM and to transfer it in a standardized format. The corresponding execution time and resource consumption for extracting the status information is lowered drastically so that it is possible to observe a very large number of devices simultaneously.

Interoperability with FDI

The FDI collaboration will enable FDI device packages to be used in both FDI and FDT/FRAME-based systems. The integration into the FDT/FRAME takes place by means of a special FDI-DTM, which interprets the components of the FDI device package. Additionally, FDT 1.x and FDT 2.0 DTMs which are not supported by the future FDI specification can be integrated into the FDT/FRAME.

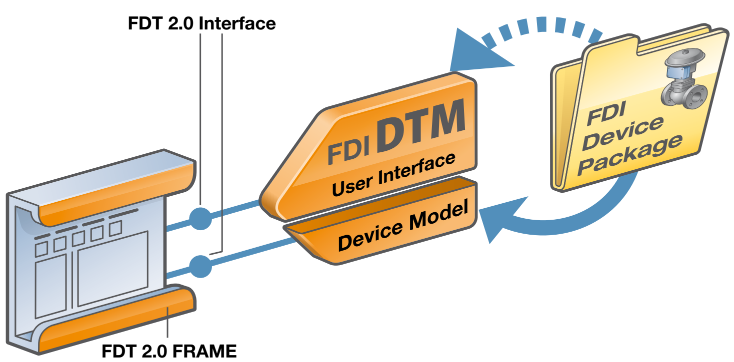


Figure 14: Interoperability between FDT 2.0 and FDI

4.4 Common Components

Interoperability

The interoperability of all the components is the most important prerequisite for the smooth function of FDT in a plant. That is why particular attention was paid to it during the development of FDT 2.0. The simplification of interfaces as well as the more detailed specifications — compared to the previous version — push in this direction. For example, the state machines of the FDT/FRAMEs and FDT/DTMs were optimized and the rules for the use of threads (executing threads parallel to each other) and their synchronization were defined precisely.

However, interoperability primarily depends on the respective products being implemented in line with specification. In order to ensure this for the most important parts of the FDT 2.0 specification, such as interface behavior, the FDT Group has declared these parts as "Common Components" that are developed and made available centrally. These components can then be licensed and used by all companies carrying out FDT projects.

During their development, new framework applications and DTMs can then use Common Components unchanged and, therefore, without the danger of individual deviations from the specification. This leads to a substantial improvement in interoperability and lowers the amount of effort needed for interoperability tests. Simultaneously, the development of DTMs and FRAMEs becomes simpler, faster and more cost-effective.

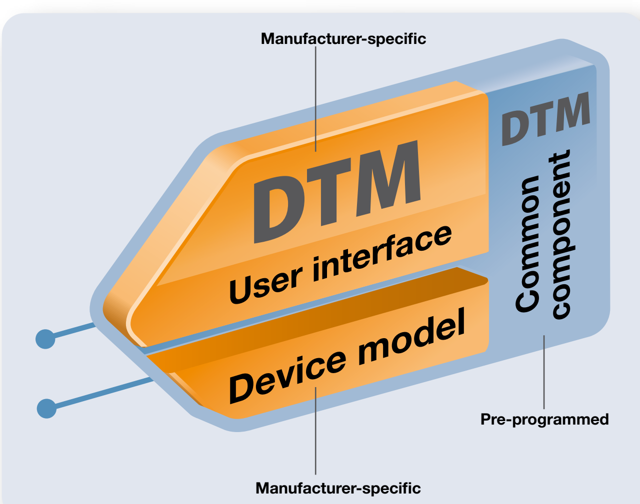


Figure 15: Common Components in a device DTM

4.5 Vertical communication

Vertical communication (nested communication) is a method by FDT to establish a targeted connection to a device within a multifaceted and hierarchical topology from an FDT/FRAME across the boundaries of various communication systems. The method is called vertical communication, since superimposed communication levels end up being bypassed during the process. The principle exists in the ongoing connection of all of the DTMs responsible for the affected access path, for which each DTM only has to support its own communication protocol without being familiar with the overall topology.

- A device DTM always needs at least one communication DTM as a channel to the first communication set-up. With appropriate gateway DTMs, communication paths can then be opened across protocol levels as well (e.g., from Ethernet over a fieldbus to HART) (see Figure 16). There are gateway DTMs for all common communication protocols; DTMs are able to be created for additional protocols and associated components. Such a structure of a layered communication path is simplified through the ability of FDT/FRAMEs to scan connected devices across protocols when a communication path is open and to be able to allocate the correct DTMs. Figure 16 illustrates vertical communication; all DTMs necessary for access to the desired component are registered in the device catalog of the FDT/FRAME. The DTMs corresponding to the access path configured in project engineering are retrieved and coupled with each other for access to the system. Figure 16 shows this mechanism based on the procedure "Load data in one device": With the call "START, write data in the device", the device data are prepared via the responsible DTM device 1, "packaged" into the HART protocol supported by this device, and transferred to the higher-level DTM 2.
- DTM 2 repackages the data/HART packet into the fieldbus protocol supported by it and passes this on to the higher-level DTM 3.

- DTM 3 extends the packaging further to the Ethernet protocol supported by it and forwards the "total packet" created by this process to DTM 4.
- DTM 4 transfers the total packet by means of a telegram structure to the controller (5) and thus to the highest level of the real physical network.
- Devices 6, 7 and 8 take over the data packet from the respective higher-level device, unpack it by one layer each, and forward it to the subordinate device.
- The chain finally ends in the device with the data loaded (END, data written into the device).

The network topology from Ethernet, fieldbus and HART is shown here only as an example. This method, also called "routing through the system topology," can be applied without constraints to other systems and specifically, systems that have a much more complex design.

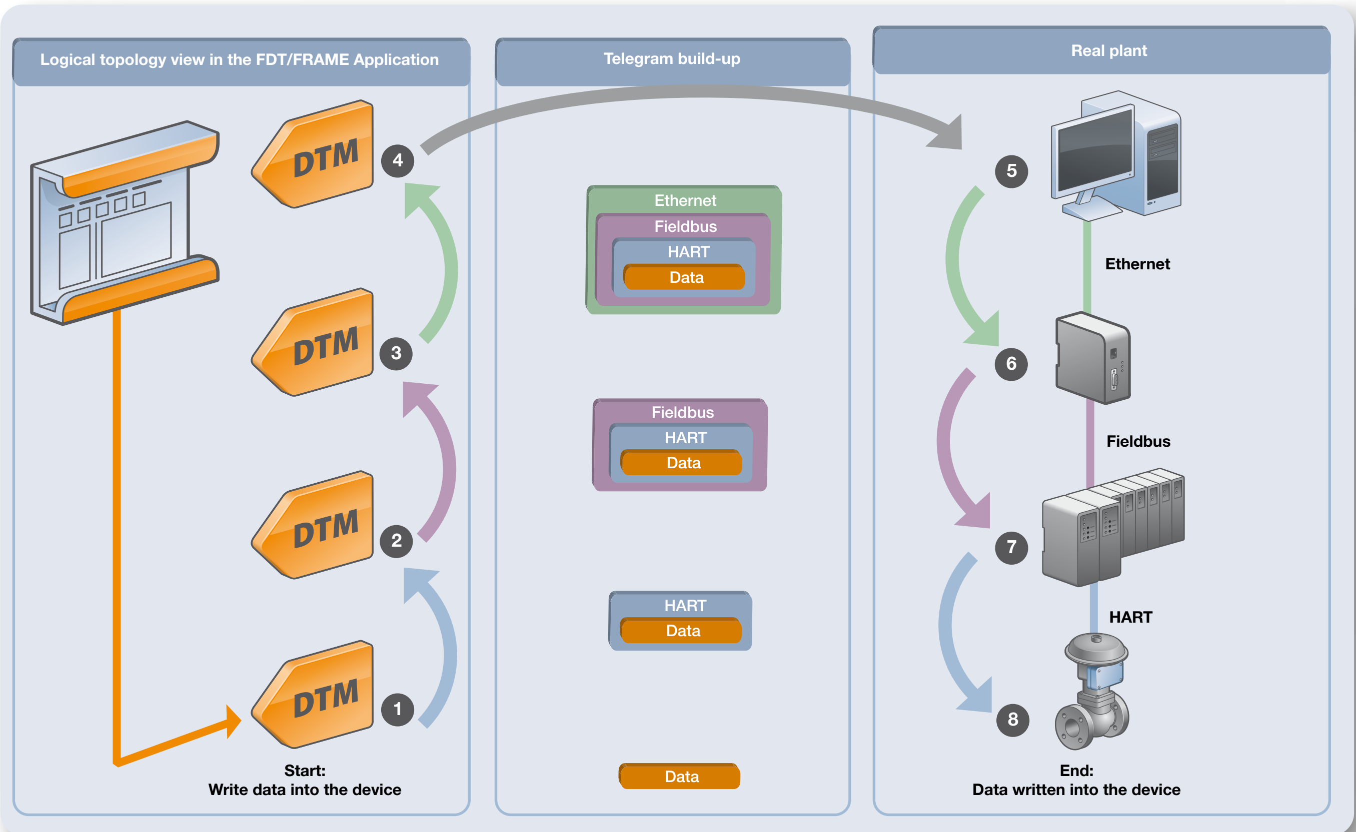


Figure 16: Vertical communication over multiple levels

4.6 PLC tool interface

Integration of process signals into PLC programming systems

FDT Technology is used primarily for the integration and configuration of field devices in controllers and control systems, or in plant asset management systems. The PLC programming system accesses devices or their DTMs via its FDT/FRAME, which is why FDT already supports — in view of the multitude of bus types in factory automation — a large number of various fieldbus and Ethernet protocols. However, a continuing user request is to take process signals into account during device integration so they can be processed by the PLC programming system for all fieldbus systems in a uniform way.

The process signals provided by the device DTMs possess a fieldbus-specific format. However, in the past, no standardized description of these signals existed in a fieldbus neutral form, with a basis that would make uniform processing possible. This requires familiarity with bus-specific formats from the PLC programming system; a substantial effort in light of the multitude of bus types.

With the help of the PLC tool interface, the process signals of the devices can be transferred to the programming system in FDT-based systems with fieldbus neutrality. Thus, a host system based on FDT technology offers the greatest scope and flexibility for device integration across all process and factory automation networks and devices.

Here, the FDT 2.0 specification relieves the PLC programming system, and thereby the user, of the task of dealing with bus-specific details. The following highlights the mechanism:

- In the fieldbus-dependent variants in use up until now, the programming system retrieved the information for the process signals directly at the device DTM (see Figure 17, on the left).
- Using new fieldbus-independent variants, the integration takes place in a two stages

First, the communication retrieves the bus-specific process signals from the device DTM (see Figure 17, on the right), since it knows their formats. The communication DTM combines this information in a fieldbus-neutral form and provides it to the programming system for retrieval.

This fieldbus-independent integration of process signals is made possible through the definition of additional FDT interfaces and data types for the communication DTM to form a neutral description of the process map. This innovation allows a consistent and uniform display of process data from different fieldbuses and the assignment of the data types to IEC61131 data types. The result is a complete system for the integration of different fieldbus systems and devices from different manufacturers. The fieldbus-independent and standardized use of I/O signals in the PLC programming system is a large step forward for both the user as well as the providers of such systems.

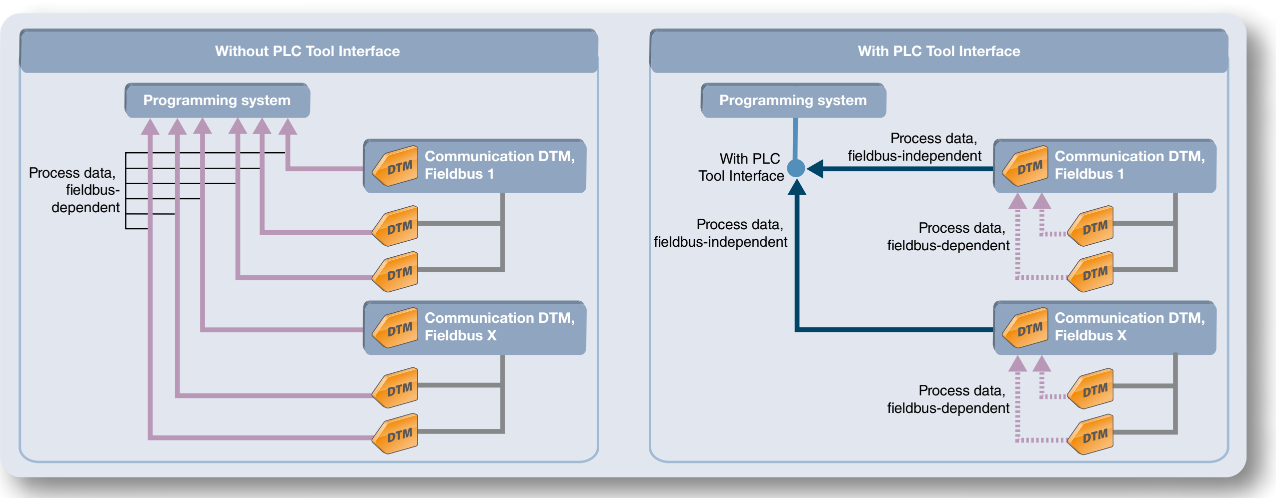


Figure 17: PLC tool interface for fieldbus-independent integration of process signals

4.7 FDT 2.0 as an object model

In object-oriented programming, "object" is understood to be a unit that can be accessed through its interfaces and is identified by certain properties while being encapsulated from external interaction. The "interfaces" in this context describe the functions and procedures in an object and the action of calling them. The properties of objects and mechanisms they use to link to each other and communicate between each other are called an object model.

FDT 2.0 is based on such an object model; the functions of FDT 2.0 are represented through objects and their interfaces, as the following sections illustrate. The four basic objects are the FDT/FRAME, the FDT/DTM business logic (BL), the DTM user interface (UI) and the communication channel. Figure 18 shows these objects together with the relationships between them. Microsoft's .NET technology is used to implement the object model.

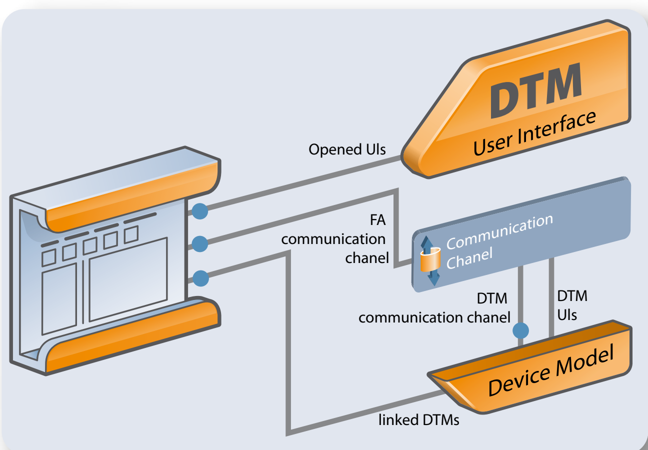


Figure 18: FDT 2.0's object model

FDT/FRAME

The FDT/FRAME object contains DTMs and provides them with a runtime environment. At the same time, it provides interfaces for interactions both from business logic or user interface to their environment as well as between each other. A direct interaction of business logic and user interface is intentionally prohibited in FDT 2.0. The FDT/FRAME itself is subdivided into.

"Business logic," which is used for running the functions like communication with devices, browsing the FDT topology, saving data or interacting with DTM UI objects, and "User interface," which establishes user access to the DTM functions (see Figure 19).

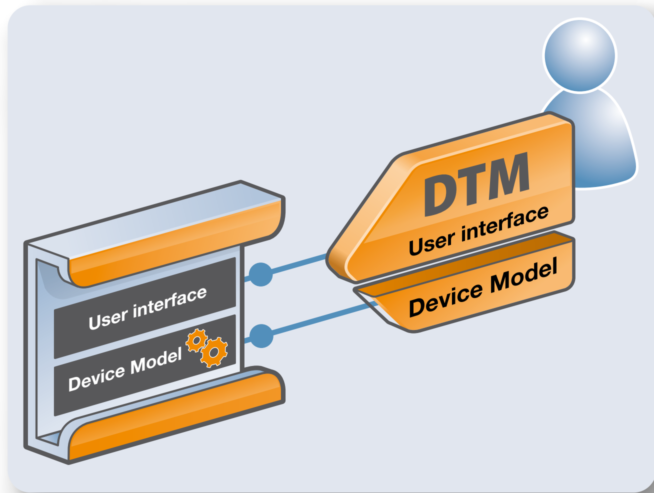


Figure 19: Object model of FDT/FRAME according to FDT 2.0

Web Services

Enabled by FDT object modeling, and coming soon to the FDT architecture, a standardized mobile access approach utilizing browsers, apps, standalone applications, or anything else capable of interfacing via web sockets, will be available. Users will be able to take advantage of standard browsers to gain access to device DTMs and FRAME-enabled systems, or write custom apps and programs. Devices utilizing Web Services can be connected through wired media (e.g., Ethernet), fiber optics, or wireless networks.

The architecture supports a robust layer of security. For example, it leverages vetted industry standards and encrypted communications, with transport layer security (TLS) utilizing secure HTTPS and WSS communication protocols. There are also 509v3 certificates for authentication and authorization of client devices. Additionally, the solution employs on-the-wire security and roll-based user security as appropriate.

Business logic

The business logic is responsible for processing data and encapsulates device-specific and protocol-specific functions, which puts the FDT/FRAME in a position to interact with any devices or protocols without knowing their specific details. In the process, the business logic and the FDT/FRAME interact via defined interfaces. Figure 20 shows the "Information objects" (info) used in the process:

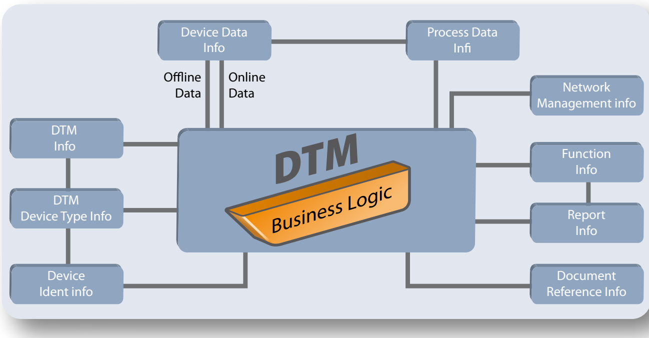


Figure 20: Information objects from DTM Business Logic

The object's **FDT/DTM, DTM device type and device identifier info** encapsulate information about the device such as type, manufacturer, hardware and software version, etc.

The object's **Device data info** encapsulates the device specific data (parameters)

The object's **Process data info** encapsulates the process-oriented information regarding the device integration such as data types, signal direction, etc.

The object's **Network management info** encapsulates information regarding the network such as bus addresses, tag numbers or bus-specific settings. The information is protocol specific.

The object's **Function info** encapsulates information about the functions of the DTM, such as identifier, status (active or not active), etc.

The object's **Report info** encapsulates information about current device data for the purposes of documentation or archiving.

The object's **Document reference info** encapsulates reference information about external documents, for displaying the references (e.g., in the FDT/FRAME).

DTM user interface

The DTM user interface object is built up using graphical control elements, which are either integrated directly into the UI portion of the FDT/FRAME or, as a proxy, access an external program made available using the DTM. The contents and design of the UI are device-specific, however, they are to follow the DTM Style Guide in appearance.

Communication channel

The "communication channel" object opens the path to a communication system and is technology-dependent. It is the access point to a standardized fieldbus system, a proprietary communication bus or a point to point connection. Communication channels can be part of both an FDT/FRAME and a DTM BL.

A communication channel possesses interfaces for establishing or disconnecting connections to a device, sending messages or scanning a plant hierarchy regarding available devices.

4.8 DTM categories

The variety of device types used in an automation system corresponds to an identical variety of DTMs, which are thus broken down into categories. The three main categories — device DTM, communication DTM and gateway DTM — are described in Chapter 3.2. In addition, the following categories are defined:

- A Composite device DTM for modularly designed devices (special gateway DTM)
- A Module DTM for the application software of a hardware module (special device DTM)

Composite device DTMs and module DTMs belong together and describe a modularly designed device.

4.9 FDT system topology

An FDT system topology results from the hierarchical structure of FDT/DTMs, for which a "parent DTM" can be connected to one or more "child DTMs." The FDT/FRAME is responsible for topology management (i.e., the design of the hierarchical structure), which also defines the communication path (routing) to a specific device in the plant. The totality of all connections between the DTMs of a topology is called FDT topology and is described in more detail in IEC 62453-2. Figure 21 shows a simple topology, in which a communication channel is being used as a connection between a communication DTM and a device DTM and is establishing access to the fieldbus at the same time. Another example is vertical communication, which is discussed in Chapter 4.5.

4.10 FDT communication

FDT communication is based on a communication request on the part of the device DTM to the communication channel. After initialization of the communication channel, the FDT/FRAME causes the device DTM to establish a connection between its device logic (business logic) and the field device (see Figure 22). If the establishment of a connection is successful, the device DTM can send additional requests to the channel, which subsequently communicates with the field device via the bus system.

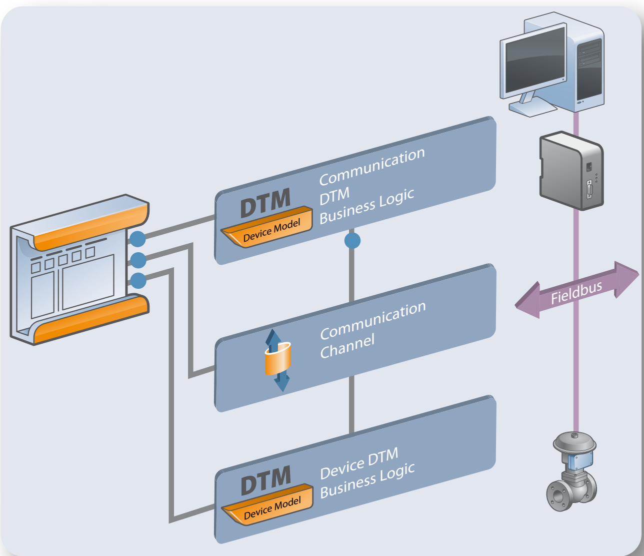


Figure 21: System topology in FDT 2.0

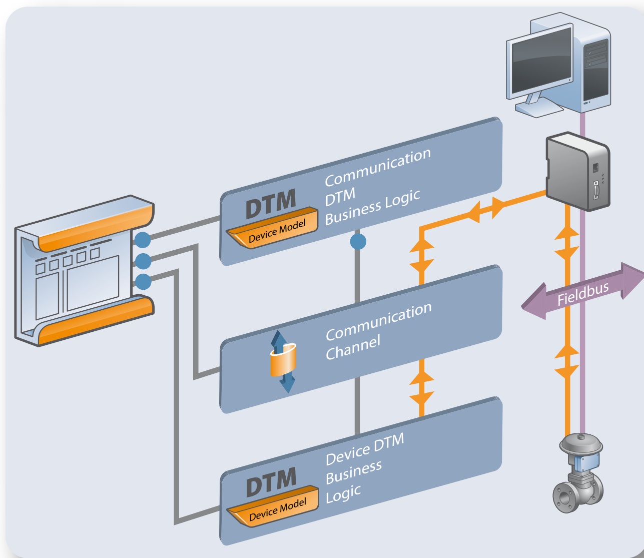


Figure 22: Communication relations for FDT 2.0

5. Quality Assurance

The FDT Group ensures the interoperability of its technology with a highly developed quality assurance concept for FDT/DTMs (see Figure 23) via the official FDT testing tool "dtmINSPECTOR." Certification ensures DTM conformance to the FDT specification, includes interoperability tests, and checks compliance with guidelines such as the DTM Style Guide.

DTM testing is available for:

- FDT/DTMs based off of FDT 1.x
- FDT/DTMs based off of FDT 2.0 (NOTE: A standard certification for FDT 2.0 DTMs is restricted to DTMs using DTM Common Components.)

Certification test for DTMs

A basic property of FDT Technology is the manufacturer-independent development of FDT/DTMs and FDT/FRAMEs and their use in customer applications. Therefore, a quality assurance concept for all FDT/DTM products (device DTM and communication DTM) is necessary to safeguard conformity with the FDT specification and integration with FDT/FRAMEs.

In 2005, the FDT Group defined an extensive testing and certification process, which encompasses the establishment of a central certification authority and multiple test labs spread across the globe along with related lab auditing and accreditation. Various test scenarios have been developed for this that contain both the sequences for DTMs as well as the results expected regarding complete conformity. In order to be able to let tests and their evaluation be performed automatically, a powerful tool named dtmINSPECTOR is used. The tool carries out the test run, in part automatically, and compares the behavior of the test pieces to the anticipated results. Finally, the tool creates a report that indicates whether or not the behavior of the FDT/DTM conforms to specifications during all test runs. If a test is false, the report also contains the precise information about the test run, which can allow simplified analysis and quick elimination of the error. The certification tool can be of great use to the DTM manufacturer even during DTM development for being able to test the implementation for conformity to the FDT specification continuously.

For example, test scenarios for multi-workstation systems and their capacity for parallel device access or even targeted faults and communication failures can be simulated and tested.

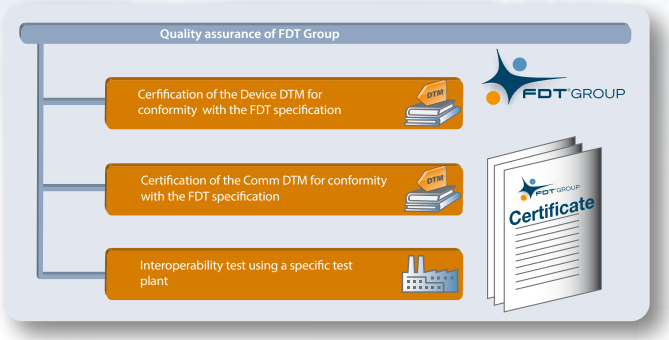


Figure 23: FDT Group quality assurance

Qualified test labs

To ensure independent FDT/DTM tests of the proper quality, test labs are accredited by the FDT Group. Prerequisites for this purpose are technical expertise, suitable metrological facilities, the ability to handle the test tools, a neutral and independent position on the market and willingness to cooperate with both the customer and FDT Group. Approval as a FDT test lab is facilitated through an independent auditor. The auditor checks for the previously mentioned requirements and, upon successful completion, gives a recommendation to the FDT Group, which then makes a final decision. A list of accredited test labs can be found on the FDT Group website (www.fdtgroup.org).

Certification process of an FDT/DTM

The FDT/DTM is sent along with the associated device to the test lab and is subjected to the official test there (see Figure 24). To do so, a plant with components that have already been certified is used. The test contents include, in addition to correct installation and uninstallation, the DTM state machine as well as the return values of the defined interfaces. The stability of the DTM is tested and, insofar as it is part of the specification, documented in the report.

The conformity of the DTM to the DTM Style Guide is also a component of the testing process (only for FDT 2.0 DTMs). In the process, the design and behavior of the DTM from a user's point of view (look and feel) are tested. This encompasses the use of certain icons, terms and formulations, which are displayed on the DTM's user interface in any of a multitude of languages. Doing so ensures a uniform appearance and uniform terminology for all DTMs in an FDT/FRAME. However, the tests do not encompass function testing, such as that of the internal DTM parameter model. For that, the DTM manufacturer must perform its own function tests.

Finally, the test lab creates a report for the DTM manufacturer, which requests the certification of its DTM at the FDT Group certification authority. After checking the report and results, the certificate is issued when the findings are positive and the DTM is added to the FDT/DTM product catalog on the FDT Group website at www.fdtgroup.org. The certificate is permanently valid for the respective version of the DTM that was tested. However, each new version must go through the described procedure again.

A test report is related to precisely one device type and one operating system. If the tested DTM supports multiple device types, the manufacturer can declare that these other device types behave exactly the same as the tested DTM. Then those device types are included in the certificate as well.

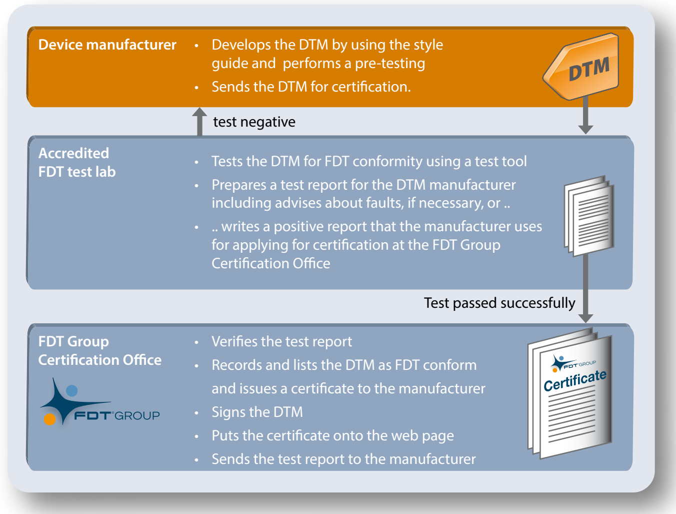


Figure 24: Sequence of the certification process

Arbitration

If the test result is negative, the DTM manufacturer has the option of lodging an objection, for example if there is suspicion of a faulty test environment. In this case, the manufacturer can appeal to the FDT Group directly, which will examine the tests and discuss the problem. The Test & Certification working group will suggest a solution that is presented to the FDT Group for a decision. FDT Group then decides whether the manufacturer's objections are justified. If the specification, testing tool or test case has an error, this is corrected as quickly as possible. The DTM receives the certificate with an official description of the exception.

Ensuring interoperability

The test tool behaves like a special FDT/FRAME for the certification test of an individual DTM. Depending on circumstances, FRAMEs such as a control system in conjunction with multiple communication DTMs, gateway DTMs and device DTMs can behave differently than in a defined test scenario. Therefore, the FDT Group operates a special test lab with various devices and FRAMEs provided by member companies. This way, developers can perform interoperability tests at any time. Additional FDT Group workshops for interoperability support DTM quality assurance in complex environments.

6. Development Resources

Specifications:

Downloads for the FDT specifications, style guide, annexes for supported communication protocols, along with the OPC UA companion specification, are available on the FDT Group website.

Download specifications by visiting: **https://fdtgroup.org/development/specifications/**

Developer Tools:

To aid manufacturers in innovative product development, the FDT Group established a set of tools saving countless engineering hours when creating FDT-enabled solutions. For example, Common Components are available for both FDT/FRAME-enabled systems and FDT/DTM device solutions to create a library of routines to simplify the development process. The official test and certification tool for DTMs, dtmINSPECTOR, is also available. These tools ensure specification compliance, greatly enhance interoperability, and work together to ease developer efforts in delivering robust FDT®-enabled solutions while accelerating time to market.

For more information, visit: **https://fdtgroup.org/development/developer-tools/**

Service Providers:

The FDT Group partners with several leading service providers that offer a wide range of assistance for companies interested in FDT®-enabled product development. They can advise about system and device integration, and help suppliers future-proof their system and device management strategy with FDT to support the new era of automation.

For more information, visit: **https://fdtgroup.org/development/service-providers/**

Integration Lab:

The FDT Group established and maintains an FDT Integration Lab designed to allow comprehensive integration focusing on interoperability testing for FDT/FRAME-enabled industrial systems and instrumentation products. Located at Dietz Automation in Neukirchen, Germany, the Lab is staffed with FDT test support experts to assist in meeting integration and interoperability testing requirements.

For more information, visit: **https://fdtgroup.org/development/integration-lab/**

Contact Us:

For questions or to receive additional information, please contact us at: **info@fdtgroup.org**

7. Glossary / Abbreviations

ActiveX	Designation for a (software) component model from Microsoft
Business logic, or domain logic	A term in software technology with the goal of delimiting the logic of a software system based on the task assignment from the technical implementation. It is also designated as the layer in the software responsible for processing data. This layer contains programs that process data from the lowest layer and prepare it for display in the presentation layer.
CIP™	Common Industrial Protocol — A category of protocols such as DeviceNet, ControlNet and Ethernet/IP.
Client/server system	Interaction of various computers in a network. In the process, client computers belonging to the network fetch certain services from a large-capacity server computer. Multiple clients can operate off of one server and a client can retrieve a service from multiple servers.
COM	Component Object Model — The component object model is Microsoft's approach to developing reusable software components. It is a binary standard that describes calling conventions in a language-neutral form.
DCS	Distributed Control System — A (distributed) process control system is part of a production system. It is used for monitoring and controlling decentralized devices including external operator interactions.
Device catalog	In FDT/FRAME applications, device catalogs are usually used that list every device type known in the system.
FDT/DTM™	Device Type Manager — A device type manager is a software component that represents a device (device driver). It contains device-specific data, functions and logic rules.
FDT	FDT is the designation of the actual standard. FDT standardizes the communication between field devices and systems. It is communication protocol-independent and non-proprietary and has established itself as an open industry standard.
FDT/FRAME™	The FDT/FRAME provides a common runtime environment and renders a graphical user interface to FDT/DTMs, and is the unifying tool embedded in system/host applications like asset management tools, Programmable Logic Controllers (PLCs) or Distributed Control Systems (DCSs). The FDT/FRAME is designed to administer entire device instances and save their data without the need to possess device-specific knowledge. This guarantees enterprise-wide, uniform lifecycle management, and supports multi-workstation and single workstation environments.
FF	Foundation Fieldbus — The Foundation Fieldbus is a 2-wire fieldbus that provides devices with current.
FITS™	FDT IIoT Server — To empower the future of IIoT and Industrie 4.0, FITS enables mobility, cloud and fog enterprise applications, as well as sensor-to-cloud and enterprise-wide connectivity. The IIoT server employs FRAME™ and DTM™ business logic at the heart of its client/server architecture and can scale to suit the needs of a single manufacturing facility or an entire industrial enterprise.
Frame	A structured packet where data is placed for transmission. The data is split into small units and provided with control system information. Each packet of this type is sent as an individual unit called a *frame*.
HART®	Highway Addressable Remote Transducer — HART is a communication protocol that is overlaid on the analog 4-20 mA process value in order to transmit extended digital information.
HMI	Human-Machine Interface — The human-machine interface or user interface is the layer between user and machine.
Instance	In object-oriented programming, an instance is an object that is created at the runtime of a class. For FDT: a DTM can be allocated (instantiated) to one or more devices.
.NET	A software platform developed by Microsoft for developing and executing programs.
.NET class	Innovation (partial classes) in the .NET framework 2.0 type concept, with which the developer can split the program code of a class over multiple individual class definitions.
PLC	Programmable logic controller — A device that switches on (sets) or switches off its outputs based on the status of the input signals. It replaces the relay circuits of sequence control.
WinForms Control	Windows Forms is one of the two desktop interface libraries in the .NET framework.
WPF	Windows Presentation Foundation (WPF) is a graphical framework and part of Microsoft's .NET framework 3.0. Offers a uniform programming model for the development of an extensive user environment for intelligent Windows clients.
XML	Extensible Markup Language — Extensible markup language is a simple, but flexible, text format for exchanging data between computer applications.

Join the growing number of leading automation companies in support of FDT

Join Today

FDT®GROUP

Empowering the Intelligent Enterprise

5 Industrieweg | 3001 Heverlee | Belgium | +32 (0) 10 22 22 51 | info@fdtgroup.org

For more information, please visit www.fdtgroup.org or contact the FDT Group Business Office:
Phone : +32 (0)10 22 22 51 · Email : businessoffice@fdtgroup.org