

TECHNOLOGIEN MIT  
VORSPRUNG



# isHRT CommDTM

User Manual



Copyright © ifak system GmbH 2002-2009, all rights reserved.

No part of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping or storage in an information retrieval system - without prior written permission of the ifak system GmbH.

Information in this manual is subject to change without notice and does not represent a commitment of the ifak system GmbH.

ifak system GmbH  
Oststr. 18  
D-39114 Magdeburg (Germany)

[www.ifak-system.com](http://www.ifak-system.com)

Tel.: +49 - 3 91 - 54 45 63 - 0  
Fax: +49 - 3 91 - 54 45 63 - 99  
Email: [fdt@ifak-system.com](mailto:fdt@ifak-system.com)

## **Chapter 1: Introduction**

Software Installation .....	1
General Remarks .....	2
Our Service .....	2

## **Chapter 2: FDT – A Brief Introduction**

FDT-Specification Basics .....	3
Object Model .....	4
Component interaction .....	7
Summary: Objects in the FDT concept .....	9
DeviceDTM .....	9
ActiveX .....	9
CommDTM .....	9
CommChannel .....	10

## **Chapter 3: Description of the isHRT CommDTM**

Dialogs .....	12
Driver Configuration .....	12
Channel Configuration .....	14
Connecting DeviceDTMs .....	16
Device List .....	17
Changing Polling Address .....	20
Changing Polling Address, DeviceDTM Address and Tag .....	21



# CHAPTER 1 :

## INTRODUCTION

### Software Installation

The operation of the software isHRT FDT requires the correct installation of the HART interface isHRT USB and isHRT USB $\epsilon$ X respectively. The driver software delivered with our interface permits the access to the HART Firmware. The interface between application and HART firmware comes in the shape of a Dynamic Link Library (DLL).

The delivered software isHRT FDT supports the operating systems Windows 2000, Vista, XP and Windows 7. It is placed on the enclosed CD. The installation follows the steps mentioned below:

- Insert the installation CD into your drive.
- Start `setup.exe` and proceed according to the instructions displayed on the screen.
- The software is installed in the default program directory of your computer under the subdirectory *"ifak system\is HRT FDT"*.

## General Remarks

This documentation contains important information for the correct operation of the software isHRT FDT. Please read this documentation carefully before installing the software.

Throughout the manual different fonts are used to indicate different meanings.

`Courier New`: file names, code sequences

`Arial`: names, commands, remarks

## Our Service

In case you have any questions concerning the software isHRT FDT which can't be solved by this manual, please contact our service by phone, fax or e-mail.

ifak system GmbH  
Oststrasse 18  
D-39114 Magdeburg (Germany)  
[www.ifak-system.com](http://www.ifak-system.com)

Tel.: +49 - 391 - 544 563 - 10

Fax: +49 - 391 - 544 563 - 99

e-mail: [fdt@ifak-system.com](mailto:fdt@ifak-system.com)

## CHAPTER 2 :

# FDT – A BRIEF INTRODUCTION

## FDT-Specification Basics

The widespread technologies **COM** (Common Object Model), **XML** (eXtensible Mark-up Language) and **ActiveX** (a COM based technology for GUI elements) form the basis platform of FDT.

COM integrates the different software components and facilitates the interaction between them. Another example for the use of COM as underlying concept of an automation standard can be found in the area of online data exchange. There, the specification OPC (DA) defines the interfaces and mechanisms for the interaction between independent software components employing COM functionality.

Compared to OPC the aims of FDT are quite more challenging. The COM functions transfer XML documents, which contain the data to be communicated between different components. The data which may be of different levels of complexity convey information about the various aspects of the software component (general information like manufacturer and device type; supported functions, communication data).

The ActiveX technology was chosen in order to integrate device specific dialog elements in the HMI of the engineering system. FDT adds new functions to the already existing ActiveX function set, to adapt the standard to the special needs of FDT. For a seamless integration of all components FDT specifies (with more or less depth) the object model, interfaces and XML schemas, as well as a session and data management.

## Object Model

The object model contains the following four elements:

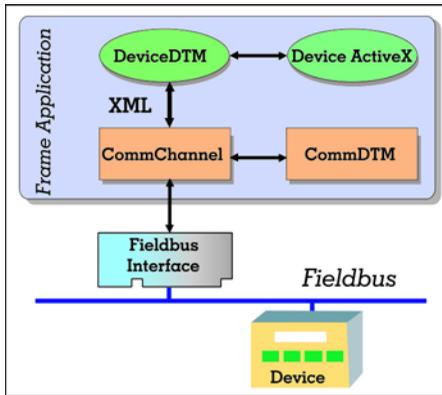
- Frame Application
- Device Type Manager (DTM)
- Channel
- ActiveX

An engineering system or a configuration software forms a **Frame Application**, which defines the run-time environment, administers the project data and stores the persistent data. In most cases also the access to the underlying fieldbus communication is provided.

A **DTM** as device data server handles the tasks of management and configuration of a device class, as defined by the manufacturer. This device class might be one device type or a group of devices. A typical DTM contains no graphical HMI. In most cases a DTM is invoked for every device.

The parameter of the device as well as the HMI elements can be influenced or activated by the user of the DTM. Most HMI elements are implemented as **ActiveX** components, although the inclusion of external programs is also allowed. The ActiveX fits seamlessly into the **Frame Application**. The user is not able to determine where the called HMI came from. It appears as a natural part of the engineering system.

Figure 1: FDT Object Model



At this point two new terms are introduced, which are not contained in the FDT specification but seem to be helpful for an easier understanding of the specification.

A DTM for a device class without any direct access to the communication (the use of the FDT communication functions notwithstanding) is named **DeviceDTM**. In order to configure a field device (sensor, actuator) within a FDT compliant engineering system one Device DTM must exist for every device. The DeviceDTM will be provided by the device manufacturer.

A DTM for a device class with direct access to a communication device (e.g. a Profibus interface) is named **CommDTM**. The CommDTM encapsulates all communication specific aspects. It is responsible for the management and configuration of the communication interface. Other applications (DTMs) call its CommChannels for field communication access. The CommChannel can be provided by an additional CommDTM or by the Frame Application.

DeviceDTM and CommDTM differ mainly in the type of channels a DTM supports. A Channel represents either a communication (CommChannel) or data channel (DataChannel). A

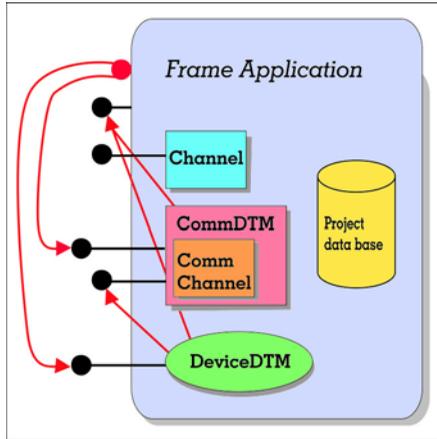
CommChannel allows other DTMs an access to the fieldbus communication. It supports the interface “IFdtCommunication“, which groups the communication access function. A DataChannel allows other DTMs an access to the data values and is provided by the device DTM. The values could be the process values of a field device, which were communicated by cyclic data exchange.

A device in FDT is managed by its dedicated DeviceDTM. Its ActiveX based HMI allows the user the access to its parameters and functions. The DeviceDTM communicates with its device through a CommChannel and the underlying fieldbus interface.

Lets draw a comparison between the FDT terms DeviceDTM, CommDTM, CommChannel and the terms in use within the domain of automation software. A DeviceDTM (and its ActiveX components) replaces the manufacturer specific field device configuration software and the CommDTM (and its ActiveX components) replaces the manufacturer specific fieldbus interface configuration software. A CommChannel equals the driver library of a communication interface.

If a DeviceDTM is called and attached to a certain device, this DeviceDTM instance is associated with a CommChannel. The device can establish a connection to the device through this CommChannel. Contrary to today’s practice the DeviceDTM can expect an interface with a manufacturer independent behaviour.

Figure 2: Frame Access



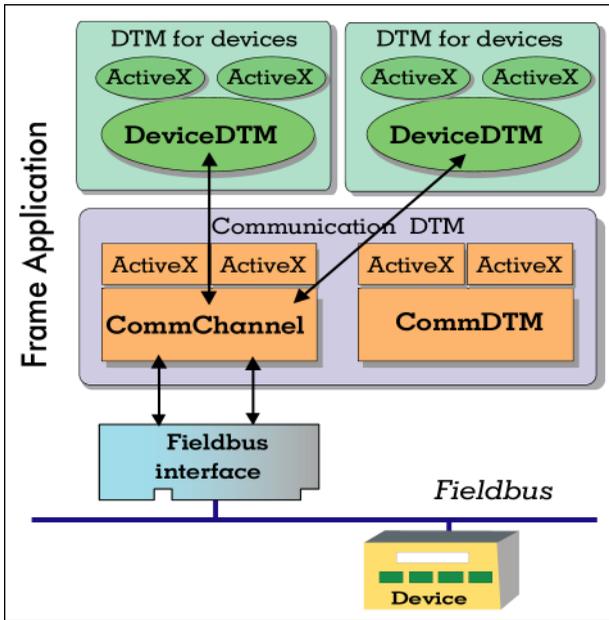
The definition of interfaces and XML schemas allows the connection and interaction of the pictured software components. About 30 data interfaces support the flow of information: A frame application accesses the DTMs through the interfaces “IDtm“ and “IDtmInformation“ in order to manage the DTM and inquire about the DTMs capabilities. The DTM itself accesses the frame through another set of interfaces which allow the DTM to acquire information from and transmit data to the frame.

## Component interaction

Before going into more detail a short example about the interaction of the different components mentioned so far might help to deepen the understanding of the matter. Let us picture a FDT based system. The run-time environment (the frame application) was started already.

In the next step we add a CommDTM to the system topology of the frame application. The frame is now able to inquire the CommDTM about the supported CommChannels.

Figure 3: Interaction of the Comm-Objects



If the check was successful, what means that the channel and the DeviceDTM speak in the same "language" (HART, Profibus DP/V1 etc.), the communication between channel and DTM can be established. To achieve this the channel supports the interface *IFdtCommunication*. The interface contains the fieldbus access functionality (connection establishment, termination, data transfer). The communication between both components is now established.

If the DeviceDTM invokes a communication request via *IFdtCommunication* the channel is responsible for the handling of the request. The response of the device is transmitted to the DTM through its *IFdtCommunicationEvents* interface.

## Summary: Objects in the FDT concept

### DeviceDTM

A DeviceDTM is a DTM for fieldbus devices. It provides all necessary functionality of the devices in software. To achieve this, it employs the communication interfaces of the CommDTM.

Within the FDT concept the DeviceDTMs form the lowest level in the system topology. They use the services of other CommDTMs to access its field device. The parameters of the device are made available to the user by a number of ActiveX dialog HMI's which can be called from the DTM.

### ActiveX

The ActiveX are the "visible" components of a FDT application (as far as device functionality is concerned). DTM and channel objects provide interfaces for the configuration, diagnosis, documentation and parameter manipulation, while the ActiveX provide an easy way of data presentation and user interaction.

### CommDTM

A CommDTM is a DTM which offers communication functionality to other DTMs. These DTMs may have been developed for a hardware (e.g. is HART USB device) or a software (e.g. is HART OPC Server) communication interface. The CommDTM itself can employ the services of another CommDTM (gateway functionality). To allow communication with its underlying hardware or software the CommDTM offers a number of communication (Comm) channels.

## CommChannel

The CommChannel is responsible for the connection to its managed hard- or software and provides the communication interface to the DeviceDTMs. Communication requests and responses are sent, received and managed by the CommChannel.

A pointer to the main interface of the channel (*IFdtCommunication*) is passed to the DeviceDTM. The channel provides the interface *IFdtChannelSubTopology*, which supports the inclusion of a DeviceDTM into the topology of the CommChannel. It also contains functions, which allow the DeviceDTM to check whether it is able to communicate with this channel (e.g. information about the supported fieldbus protocol).

## CHAPTER 3 :

# DESCRIPTION OF THE ISHRT COMMDTM

The isHRT CommDTM is a CommDTM for the interfaces isHRT USB and isHRT USBcX. The DTM offers communication access via the isHRT Channel as explained earlier.

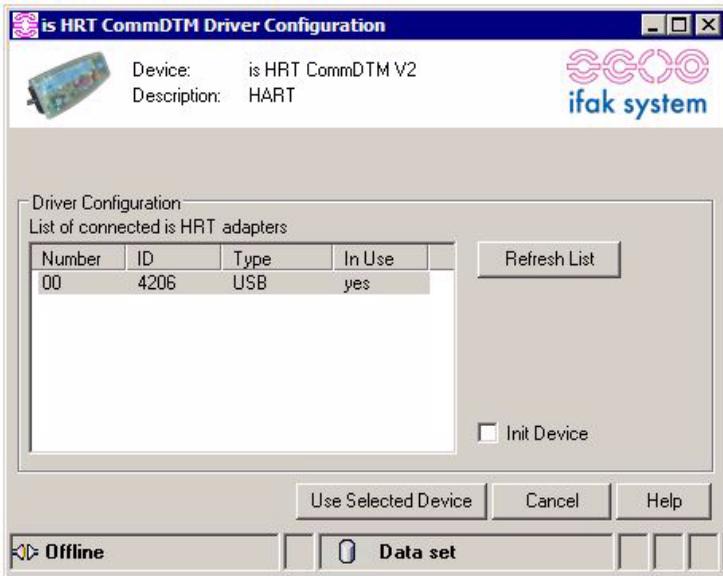
This chapter describes the configuration of the isHRT CommDTM. The configuration can be accessed by selecting the function `Channel Configuration` offered by the isHRT CommDTM.

The configuration dialog is only available in **offline mode**, i.e. the DTM is not connected.

## Dialogs

### Driver Configuration

Figure 4: Driver Configuration



This dialog is used to select one of the connected HART communication interfaces of ifak system.

- **List of connected is HRT devices**  
The table shows the properties of the found isHRT devices (HART communication interfaces).
- **Refresh List**  
Refresh the list of devices.
- **Init Device**  
If you check this field the selected isHRT device is initialized

(the green LED of the device lights). This is useful to find the selected isHRT device if more than one devices are connected.

- **Use Selected Device**

This button closes the dialog. The selected device is used for communication.

- **Cancel**

Close the dialog and discard changes.

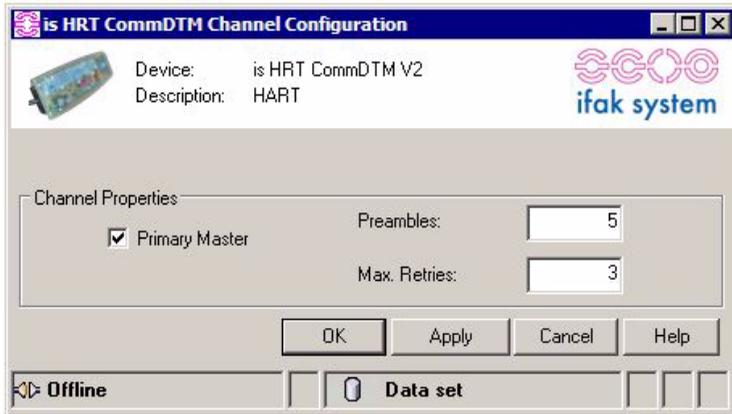
- **Help**

Display help for this dialog

## Channel Configuration

The dialog is used to configure the channel. The following options are available.

Figure 5: Channel Configuration



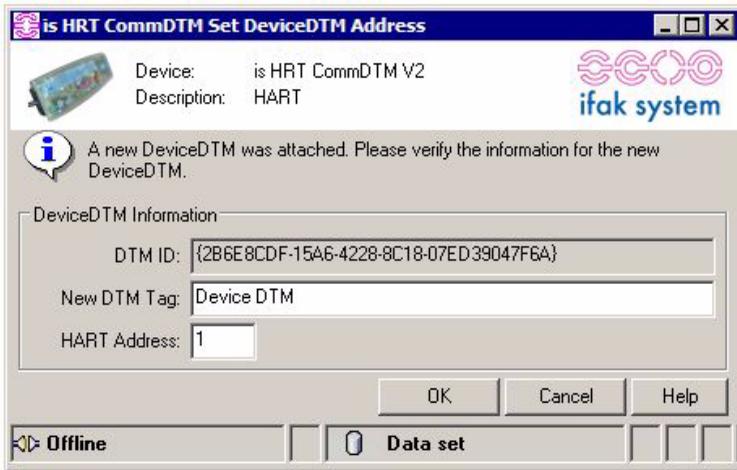
- **Primary Master**  
The parameter Primary Master specifies whether the HART master operates as the primary master or as a secondary master on the bus.
- **Preambles**  
The default setting for the number of preambles for communication with HART bus devices. This value is used to establish the initial connection to a HART device. The number of preambles to use for communication with the device is read from the device itself and used for communication afterwards.
- **Max. Retries**  
This parameter specifies how often the HART master retries to transmit information in case of an error.
- **OK**  
Apply the changed values in the DTM dataset and close the ActiveX.

- **Apply**  
Apply changed values and leave the ActiveX open.
- **Cancel**  
Close the ActiveX and discard changes.
- **Help**  
Display help for this dialog.

## Connecting DeviceDTMs

When a DeviceDTM is connected to the isHRT CommDTM then an ActiveX is opened that makes it possible to set the tag and the slave address of the DeviceDTM.

Figure 6: Set DeviceDTM Address



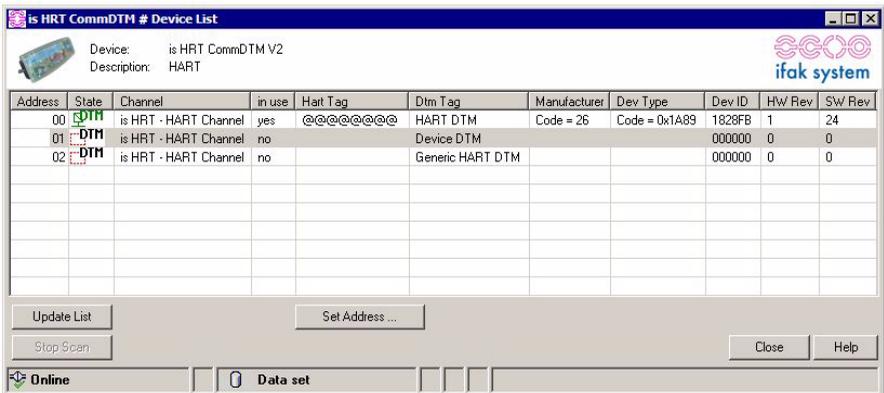
- **DTM ID**  
This is the unique ID of the DeviceDTM from the frame application. It may vary among different frame applications.
- **New Tag**  
This parameter is set as new tag in the DeviceDTM.
- **New Address**  
This is the address of the device that the DeviceDTM should communicate with. This number is the poll address. The long address with device and manufacturer ID is acquired from the device when the connection is established.
- **OK**  
Set the new values for tag and address in the DeviceDTM and close the ActiveX.
- **Cancel**  
Close the ActiveX and discard changes.

## Device List

The isHRT CommDTM offers in addition to the function `ScanRequest` an own ActiveX to scan the HART bus for devices.

This ActiveX provides information about HART devices (vendor, type, channel information) that are connected to the HART networks. If this ActiveX is called in an offline situation, it provides a list of Child-DTMs, all with no associated device.

Figure 7: Device List



The list of connected devices displays the essential information about the devices on the HART bus. This information has the following meaning:

- **Address**  
The poll address of the device.
- **State**  
The State field is used to display an icon, that reflects the state of device configuration. The following icons are possible.

**DTM**

- This entry represents a Child-DTM. This DTM is not associated to any existing HART device.

**DTM**

- This entry represents an existing HART device. No Child-DTM is configured for this device.

**DTM**

- This entry represents a HART device which is configured with a Child-DTM.

- **Channel**

This is the used channel of the isHRT CommDTM.

- **In Use**

The DTM/device is used in a communication connection.

- **Hart Tag**

The tag of the device stored in the device itself.

- **Dtm Tag**

The tag of the DeviceDTM.

- **Manufacturer**

The manufacturer of the device.

- **Dev Type**

The device type of the HART device.

- **Dev ID**

The device ID of the HART device.

- **HW Rev**

The hardware revision of the device.

- **SW Rev**

The software revision of the device.

The **Update List** button starts a new scan for devices on the HART bus. The list will be dynamically updated during the scan whenever a device is found.

During the scan which usually takes some time to complete the button **Stop Scan** is activated. It can be used to stop the scan.

The button **Set Address** is used to open the ActiveX to change the address fo the selected DTM and/or slave device.

- **Close**  
Close the Device List ActiveX.
- **Help**  
Display help for this dialog.

## Changing Polling Address

With the help of this ActiveX the polling address for an interface device can be set to a new value.

Figure 8: Change Address

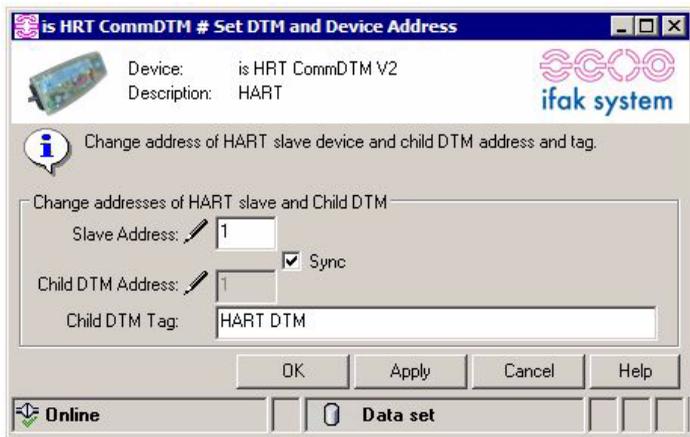


- **Source address**  
This is the current address of the device on the HART bus.  
The content of this field is set automatically when the ActiveX is opened.
- **Target address**  
This is the desired new address for the device.
- **Change address**  
Pressing this button starts the address change.
- **Status**  
Displays the status of the address change.
- **Close**  
Close the ActiveX without any further changes.
- **Help**  
Display help for this dialog.

## Changing Polling Address, DeviceDTM Address and Tag

With the help of this ActiveX it is possible to change the address of a HART slave device together with the address of a connected DeviceDTM. This ActiveX is opened by the Set Address button in the device list ActiveX for list entries that represent devices with assigned DTMs.

Figure 9: Set DTM and HART Slave Address



- **Slave address**  
The new address for the HART slave device.
- **Sync**  
Keep the values in the slave address and DTM address fields the same.
- **Child DTM address**  
The new address for the Device DTM.
- **Child DTM Tag**  
The tag of the Device DTM
- **OK**  
Change the address/tag as selected and close the ActiveX.

- **Apply**  
Apply the address/tag changes and leave the ActiveX open.
- **Cancel**  
Close the ActiveX and discard changes.
- **Help**  
Display help for this dialog.

Note:

---

The new device address must not be in use and no DeviceDTM may have open connections to the source address. The is HRT CommDTM checks these requirements before the address change is attempted. If any of these requirements is not fulfilled or the change of address itself fails for other reasons, then an error message is displayed in the status field.

# INDEX

## A

ActiveX 4, 9

## C

Changing the Poll Address 20, 21

CommChannel 10

Connecting DeviceDTMs 16

## D

Device List 17

DTM 4

    CommDTM 5, 9

    DeviceDTM 5, 9

## F

Frame Application 4

## O

Object Model 4

