AMS Device Manager Toolkit for HART Devices, Release 13.0

Guide to Customizing AMS Device Manager for HART® Devices

EMERSON™

Disclaimer

Copyright and Trademark Information

Contents

Appendices and reference

1 Overview of AMS Device Manager-related toolkits

Starting with AMS Device Manager 10.5, any DD registered at the HART Communication Foundation (HCF) will load and run on AMS Device Manager. This manual is for use in supporting your device on older versions of AMS Device Manager or for taking advantage of extra features available in AMS Device Manager 13.0 and higher. For older versions, see previous toolkits.

1.1 What is AMS Device Manager?

AMS Device Manager is plant asset management software from Emerson for managing field instruments and digital valve controllers. Its comprehensive set of analysis and reporting tools provides a single application for predictive diagnostics, documentation, calibration management, and device configuration. Using AMS Device Manager gives you better visibility into the devices in the plant, resulting in faster startup and increased availability through more cost-effective maintenance and improved device performance.

AMS Device Manager: An Integrated Solution for Asset Management

AMS Device Manager provides an integrated operating environment that leverages the full capabilities of intelligent FOUNDATION fieldbus, HART, and PROFIBUS devices, including embedded data trending, charting and graphical display capabilities provided by enhanced EDDL technology. For a description of the enhanced EDDL features in the current version of AMS Device Manager, refer to Technical Paper 925 included with this toolkit. AMS Device Manager is based on open communication standards, and is a core component of the PlantWeb digital plant architecture.

1.2 Using AMS Device Manager

AMS Device Manager is installed on a PC as either a Server Plus station or a Client SC station. A single station system consists of a Server Plus station with no Client SC stations connected to it. A distributed system is a Server Plus Station with one or more Client SC stations connected to it, where all stations can view and access devices connected to other stations. AMS Device Manager also supports installation on compatible DeltaV workstations, providing DeltaV users with a consolidated, seamless user interface to all of their online HART, FOUNDATION fieldbus, *Wireless*HART, PROFIBUS, and conventional devices.

AMS Device Manager workstations communicate with online hard-wired and wireless field devices using one or more host system interface, including the DeltaV or Ovation host system interface, the HART multiplexer system interface, the Rosemount 3420 High-Speed Ethernet (HSE) interface, the Smart Wireless system interface, and many others. Multiple host system interfaces may be used with a single AMS Device Manager system.

1.3 AMS Device Manager functions

AMS Device Manager lets you do the following:

- Display and modify device configurations. Compare configurations and transfer values from one configuration to another.
- Record and track events, using the optional Audit Trail. AMS Device Manager records events such as device configuration changes, username administration, status alert events, test/calibration activities, and database maintenance activities.
- Display and track alerts using Alert Monitor. Use Alert Monitor to notify you of problems with devices. Alert Monitor supports alerts for HART devices, fieldbus devices, and SNAP-ON applications. Each AMS Device Manager station can see the alerts for the entire distributed system.
- View device status conditions and diagnostics. With AMS Device Manager, you can view this information for devices connected to online systems such as DeltaV or Ovation. For fieldbus devices that support PlantWeb Alerts, you can view the alerts according to category and suppress or unsuppress individual alerts.
- View process variables for HART devices.
- Perform loop tests and self-tests, and automatically document the test results.
- Streamline calibration procedures for as found and as left tests.
- Upload and download configurations to and from a Field Communicator or AMS Trex Device Communicator.
- Enhance AMS Device Manager functionality with value-added SNAP-ON applications, such as the AMS ValveLink SNAP-ON application.
- Import and export information between AMS Device Manager systems, import data from other plant asset management systems, and export AMS Device Manager data to XML files.
- Transfer and configure multiple devices with automated tools.
- Expose live device data by means of the OPC interface.

1.4 Development tools

The AMS Device Manager toolkits provide the specific tools required by developers to integrate a field device into AMS Device Manager, and to provide additional device-specific functionality through AMS Device Manager. Currently, these toolkits include:

- The *AMS Device Manager Toolkit for HART Devices*
- The *AMS Device Manager Toolkit for FOUNDATION Fieldbus Devices*
- The *AMS Device Manager Toolkit for Profibus Devices*
- The *AMS Device Manager Toolkit for HART SNAP-ON Applications*

AMS Device Manager Toolkit for HART Devices

The development tools in the *AMS Device Manager Toolkit for HART Devices* include:

- This document which includes step-by-step procedures so that a HART device will work with AMS Device Manager
- Map File Specification (Condensation of AMS-SRS-62)—A guide to creating map files for use with the Calibration Assistant SNAP-ON Application
- Map File Maker and Tester. Creates and tests map file for basic devices.

AMS Device Manager Toolkit for FOUNDATION Fieldbus Devices

The development tools in the *AMS Device Manager Toolkit for FOUNDATION Fieldbus Devices* include the *Guide to Customizing AMS Device Manager for FOUNDATION Fieldbus Devices* document that contains step-by-step procedures to ensure that a device will work with AMS Device Manager.

AMS Device Manager Toolkit for Profibus Devices

This toolkit is used to integrate an existing device description for a Profibus device with AMS Device Manager.

AMS Device Manager Toolkit for HART SNAP-ON Applications

Further functionality is achievable with HART devices integrated in AMS Device Manager through the use of SNAP-ON applications. These device-specific applications are layered on the core AMS Device Manager platform to perform additional functions for a connected HART device. The tools required to develop and test a SNAP-ON application are available in the *AMS Device Manager Toolkit for HART SNAP-ON Applications*. These tools include:

- AMS Device Manager Add-in Utility that teaches AMS Device Manager to recognize new SNAP-ON applications.
- Ancillary files and AMS Device Manager demonstration software.

Familiarity with the AMS Device Manager OPC Server architecture is required for writing SNAP-ON applications.

Note

For more information about any of these toolkits, send email to amsdevicemanagertoolkit@emerson.com

1.5 Prerequisites for using the AMS Device Manager toolkits

To be able to effectively use the AMS Device Manager toolkits to integrate HART devices, developers must have the knowledge described in the following paragraphs.

Required knowledge

Developers must understand how to create a Device Description (DD) in the Electronic Device Description Language (EDDL). For more information about DDs and EDDL, refer to the FieldComm Group's Device Description Language Specification and associated user workshops.

Other things a developer should know

For basic information on installing, starting up, and using AMS Device Manager, refer to the *AMS Device Manager Installation Guide* and Books Online. (Following AMS Device Manager installation, these are available by selecting Start > Programs > AMS Device Manager > Help from the Windows taskbar.)

Device directory path

Devices are installed in the C:\AMS\DEVICES directory.

The device installation kit will install the device in the appropriate subdirectory. In this document, the path to the DEVICES directory is represented as "...\DEVICES".

Subdirectories consist of folders with device protocol, manufacturer ID (in hexadecimal) and device type (in hexadecimal).

1.6 Where to find more information

The following table summarizes the reference material available for AMS Device Manager device developers.

Reference	Source	Address	Phone/Fax	Web Address
AMS Device Manager Toolkit for HART® Devices	Emerson	12001 Technology Drive Eden Prairie, MN 55344	P:1-800-833-8314	www.emerson.com/ams
AMS Device Manager Installation Guide	Emerson	12001 Technology Drive Eden Prairie, MN 55344	P:1-800-833-8314	www.emerson.com/ams
HART Device Description Integrated Development Environment (DD-IDE)	FieldComm Group	Re: Documentation 9430 Research Blvd. Suite I-120 Austin, TX 78759	P: 1-512-792-2300	fieldcommgroup.org

1.7 Technical support for AMS Device Manager

For technical assistance in using the AMS Device Manager Toolkits or the AMS Device Manager sample code contained on the Toolkit media, or to have a HART DD tokenized for use with AMS Device Manager, contact your Emerson representative or Local Emerson Business Partner, call 1-800-833-8314 or 1-512-832-3774, or email *amsdevicemanagertoolkit@emerson.com*.

1.8 AMS Device Manager-related abbreviations and acronyms

DD	Device Description
EDDL	Electronic Device Description Language
HART	Highway Addressable Remote Transducer

Note

A glossary of terms used in AMS Device Manager can be found in Books Online.

---

2 Introduction

This chapter introduces the *AMS Toolkit for HART Devices* and this guide. The chapter:

- Explains what is in the toolkit and who should use it.
- Briefly describes the AMS Device Manager system components and how you can extend the AMS Device Manager functionality.
- Describes the purpose of this document, how it is organized, and the conventions used in the document.
- Directs you to the next steps in customizing the AMS Device Manager application.

The parts of the toolkit that you will use to develop your devices is determined by the version of AMS Device Manager you are targeting. Starting with AMS Device Manager 10.0, if you have an enhanced EDDL with the PC host root menus defined, no other work is needed for basic device communication and configuration functionality. If you wish to take advantage of additional AMS Device Manager functionality or create an easy install kit for your customers, the following additional options are required (Not all features are required. Different versions of AMS Device Manager have different requirements.):

DD	The most basic option is an Enhanced EDDL DD with PC-based menus defined. The root menus are process_variables_root_menu, device_root_menu, and diagnostic_root_menu. If these menus are not defined, a device with just root_menu will work. The Process Variable and Status/Device Diagnostics screens will default to a minimal view.
Merge File (Required)	The minimum information in this file defines the name of your company and device to the system. Emerson also recommends you provide a template for your device. See *Create a Template and Merge File*.
DDINSTAL.INI File (Required)	This file provides the installation instructions for your device.
XXYY.INI File (Required)	This file is used to control AMS Device Manager options.
Device Calibration/ Verification (Optional)	The map file enables this functionality within AMS Device Manager. Without this file, your device will be calibrated or validated with 0 to 100 percent in and out.
Alert File (Required)	This auto-generated file enables functionality in Alert Monitor, alert simplification, categorization, and consistency across devices supported in AMS Device Manager.
Alarm File (Required)	This auto-generated file enables alert functionality in DeltaV.
FHX File (Required)	This auto-generated file enables alert functionality in DeltaV.
Help File (Optional)	Provides context-sensitive help for your device beyond DD help.

PDF Manual (Optional)	A PDF user manual can be provided for your device. These files will be launched from the Help from the right-click context menu.

2.1 What is in this toolkit

This toolkit includes documents and associated electronic files, which together provide the software development and support services tools you need to integrate a HART device into the AMS Device Manager application.

Toolkit documents

The toolkit documents include:

- This guide, which provides step-by-step procedures for field device developers to follow in preparation for integrating a field device into the AMS Device Manager application. It gives instructions for creating a Device Installation Kit and verifying the device interface.
- Map File Specification - A guide to creating map files for use with the Calibration Assistant SNAP-ON Application.
- TP-925 - Details an Enhanced EDDL functionality supported in each release.

Version compatibility

In general, a Device Installation Kit that works for a version of AMS Device Manager will also work on newer versions. It may not work on older versions. You need to target the oldest version you intend to support in your initial development.

Toolkit files

The toolkit electronic files on the Internet (*http://www2.emersonprocess.com/en-us/brands/amssuite/amsdevicemanager/Pages/Developertoolkits.aspx*) help make your development tasks easier by providing resources such as precoded functions and libraries, thus alleviating many of the labor-intensive development tasks. These files include:

- AMS Device Manager standard and interoperable DD files
- Sample DD
- MapFile Maker and Tester

Information about these toolkit files and how to use them is provided in the remainder of this document.

Online Help development guidelines

You should provide online Help for devices integrated into the AMS Device Manager application. Refer to *Appendix A* for a description of the steps involved.

This document contains specifications for creating online Help systems for AMS Device Manager device applications. It is designed to assist experienced Help developers who are responsible for creating Help systems that run in the AMS Device Manager environment.

Note

These guidelines are only for AMS Device Manager Help systems; they are not intended to be a detailed guide to the Help development process.

Creating a map file for your device

One of the optional features of AMS Device Manager is the Calibration Assistant SNAP-ON Application. In order to implement the uploading and downloading of information to/from intelligent calibrators, it is necessary to present the information in a common format. The majority of the information can be found in the HART DD for a device. Because the HART specification does not require that all parameters be in a common format, AMS Device Manager has implemented functionality that reads a .MAP file created for each device and puts the device parameters into the common format needed to manage calibration operations in your plant.

For your device to operate successfully with the AMS Device Manager Calibration Assistant SNAP-ON Application, it is necessary for you to develop a map file. To develop this file, please refer to the MAP File Specification included with this toolkit, and use the MapFile Maker and Tester applications provided.

If you are developing a new Device Installation Kit to be integrated into AMS Device Manager, you must develop, execute, and approve a test to determine if the quality of the DD are acceptable for release to production. Similarly, if you subsequently modify your DD you must repeat the testing and provide approval for release to AMS Device Manager production.

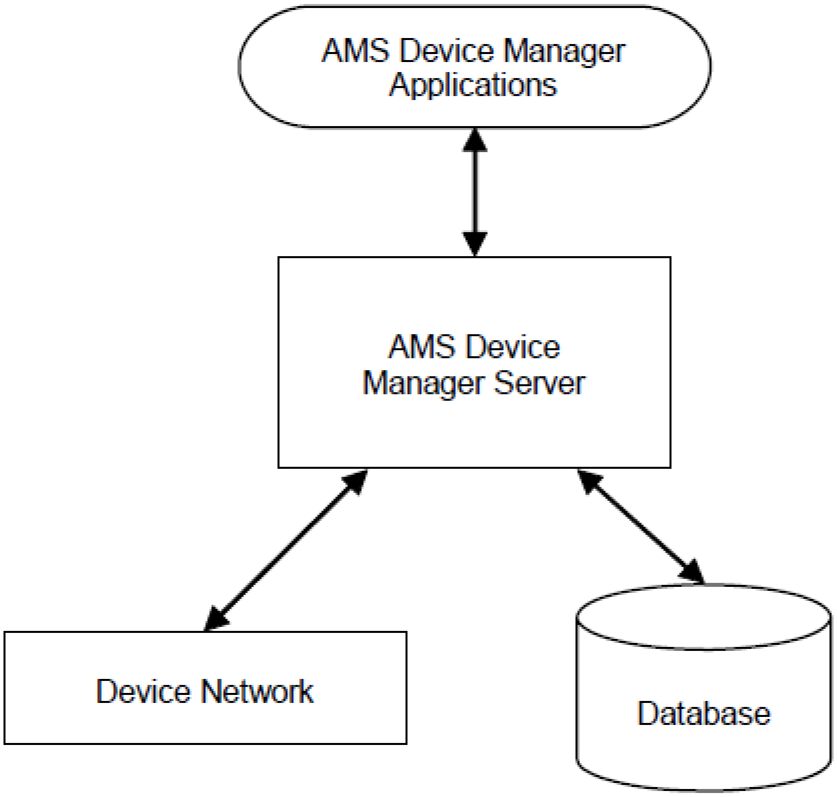
2.2 Who should use this toolkit?

The AMS Toolkit for HART Device Developers was developed for and is intended specifically for use by the HART device developer who wants to integrate a HART device into AMS Device Manager. The level of information provided in this toolkit assumes that the user understands how to write DDs in the DDL.

This toolkit does not provide instruction in programming, creating DDs, or tokenizing. If you need reference material on those subjects, refer to the HART Device Description Integrated Development Environment (DD-IDE).

2.3 AMS Device Manager architecture

The following diagram is a high-level organizational view of the AMS Device Manager system components.



AMS Device Manager applications communicate with the servers, which in turn communicate with the device network and the database. Users conduct all device management tasks through the various AMS Device Manager applications.

Access to the device network and the database information is managed by the AMS Device Manager Server (plant server and file server). Because AMS Device Manager uses a client-server architecture, applications do not have to physically reside on the same computer as the servers. This allows you to run AMS Device Manager applications (clients) on machines remote from the servers over a TCP/IP network.

The device network in the diagram represents the actual field devices.

The database is the repository for historical device data and device configuration information.

You can integrate a smart field device into the AMS Device Manager application by adding its DD and, if necessary, generating the supplemental files. This document explains how to do this and provides sample files to assist you.

2.4 Purpose of this document

This document provides a detailed, step-by-step guide for HART device developers to follow in customizing AMS Device Manager in order to integrate a field device into the AMS Device Manager interface. It does not explain the basic functions of AMS Device Manager or the details behind the AMS Device Manager interfaces. For more information about basic AMS Device Manager functionality, refer to the *AMS Device Manager Installation Guide* or Books Online.

What you should know

This document assumes that you understand how to use the features and functions of Windows operating systems. If you need more information about these operating systems, refer to the documentation that came with the systems.

2.5 Organization

This remainder of this document consists of the following sections:

- *Chapter 3* presents an overview of the procedures to follow when customizing AMS Device Manager for a device, and introduces the supporting files used in the procedures
- *Chapter 4* explains how to modify files and implement an installation program that allows your device to be installed into AMS Device Manager.
- *Chapter 5* explains how to install your device into an existing AMS Device Manager system and provides information on how to test it using AMS Device Manager.
- *Chapter 6* explains how to integrate your device into AMS Device Manager once the installation kit has been created.
- *Appendix A* describes the procedure for linking device Help to context-sensitive fields in the .ini file.
- *Appendix B* explains the procedure for generating simulation files for AMS Device Manager.

2.6 Conventions

This document uses the typographic conventions described below.

Keys on the keyboard

Keyboard keys are shown in ALL CAPITAL letters:

> ENTER

Key combinations are shown using the plus (+) sign:

> SHIFT + F1 indicates that you should press and hold down the SHIFT key, press F1, then release both keys.

Key sequences are shown using a comma (,):

> ESC, F1 indicates you should press ESC, release it, then press F1.

Filenames and directories

All files and filenames in this document use MS-DOS or Windows naming conventions. Filenames and directories are represented in ALL CAPITAL letters:

> C:\WINDOWS\FMS.INI refers to the file FMS.INI, located in the WINDOWS directory on the C drive.

Commands

Commands you must enter are shown in **bold** text, within the normal text margins:

> **DIR C:\AMSBLD\RELEASE\*.*** means you should type the command shown in the example, letter for letter.
>
> The phrase “Enter the following command:” means that you should type the command shown, then press the RETURN or ENTER key on the keyboard.

Variables

Variables in commands are shown in *italicized* text:

> COPY *source destination* means you should enter the command “copy,” but replace source and destination with your specific information.
>
> Where necessary, the variables are defined in the text.

Titles of other publications

Titles of other publications appear in *italicized* text, with the publication number in parenthesis beside the title:

> *Device Description Language Specification* (HCF_SPEC-500)

2.7 What's next?

Chapter 3 will guide you through the development process step-by-step.

Integrating a field device into AMS Device Manager

To successfully integrate your device into the AMS Device Manager application, you must thoroughly understand your device DD.

If you need to install the AMS Device Manager software, refer to the AMS Device Manager Installation Guide. To verify your device interface with AMS Device Manager, refer to *Chapter 5*.

Purchasing AMS Device Manager

You must purchase an AMS Device Manager application package so that you can effectively test the DDs you create. To purchase a copy of the application, contact your Emerson Sales/Service Office.

3 Getting started

This chapter provides an overview of the customization procedures and introduces the files you will be working with.

3.1 Overview of procedures

This guide assumes that you already have a tokenized DD file.

Customizing AMS Device Manager for a field device: overview of steps

The following general steps are necessary to customize AMS Device Manager for a particular field device:

1. Create a Device Installation Kit for installing your device into an existing AMS Device Manager system (see *Chapter 4*). You must also test the installation kit itself.
2. Back up your AMS Device Manager database.
3. Optional: create a map file.
4. Optional: create a template/mrg file.
5. Optional: create and integrate AMS Device Manager-compatible online Help files for your device. For further information on how to create and integrate these files, see *Appendix A*.

3.2 File types needed for customizing AMS Device Manager

In the course of customizing AMS Device Manager for a device, you may need the types of files listed below. You will need to create some, modify some, and simply know about others, as noted in their descriptions. If you are developing device files using enhanced EDDL, some of the files are optional.

The Toolkit media includes actual files or samples of the files you need to work with in customizing AMS Device Manager for a HART device.

Note

DEVICE, as part of a file name, refers to the base name of the device (usually the same as the base name of the DDL file).

***DEVICE*.DDL** (Optional)	The DD file contains information that describes your device to AMS Device Manager, including the device parameters and device methods.

XXYY.FM6 or XXYY.FM8 and XXYY.SYM (Required)	These files are generated as output by the DD tokenizer (XX is the device revision and YY is the DD revision). You will need these files when you install your device into AMS Device Manager. The tokenizer version determines whether an .FM6 or FM8 file is created.
DEVICE_ICON (Optional)	For an EDD, AMS Device Manager will use an image called device_icon for the image of the device. It is highly recommended to provide an icon, as this will prevent the device from using the generic icon. AMS Device Manager will scale to a 32 by 32 or 16 by 16 image for use as an icon. The recommended icon size is 32 by 32 pixels. Only JPG, PNG and GIF image files are supported. For more details on how to integrate a customized icon, please refer to TP 925.

FMS.INI (Know about)	The FMS.INI file is the AMS Device Manager initialization file. Your installation kit modifies it to indicate the tokenized device file, *XXYY*.FM8 or *XXYY*.FM6, when you start the AMS Device Manager application to test your device.
DDINSTAL.INI (Required)	The DDINSTAL.INI is the initialization file for the device installation portion of AMS Device Manager. You must modify this file to reflect your device.
.MAP (Optional)	This file contains mappings from device-specific parameters to a common format used by calibrators and export/reporting applications.
.MRG (Required)	The .MRG file is the database merge file you create that contains your device template. (You create a device template as part of the process of building a Device Installation Kit.)
Alert File (Required)	This auto-generated file enables functionality in Alert Monitor, alert simplification, categorization, and consistency across devices supported in AMS Device Manager.
Alarm File (Required)	This auto-generated file enables alert functionality in DeltaV.
FHX File (Required)	This auto-generated file enables alert functionality in DeltaV.

4 Creating a Device Installation Kit

This chapter explains how to create an installation kit that allows you to install your device into an existing AMS Device Manager system. You also need to create this installation kit to test your device interface.

4.1 Files and directories you will use

The following files and directory structures are used to add your device to AMS Device Manager. Some are required, others are optional, and some you simply need to know about.

***DEVICE*.DDL (Optional)**	These files are used as input to the tokenizer to generate the DEVICE.FM8 or DEVICE.FM6, and DEVICE.SYM files.
***DEVICE*.FMS or *DEVICE*.FM6, *DEVICE*.FM8, and *DEVICE*.SYM (Required)**	These files are generated as output by the DD tokenizer.
***DEVICE*.MAP (Optional)**	This file contains mappings from device-specific parameters to a common format used by calibrators and export/reporting applications.
FMS.INI (Know about)	The FMS.INI file is the AMS Device Manager initialization file. Your installation kit modifies it to indicate the presence of your DEVICE.DLL and the tokenized device file, XXYY.FM8 or XXYY.FM6, when you start the AMS Device Manager application to test your device.
DDINSTAL.INI (Required)	DDINSTAL.INI is the initialization file for the device installation portion of AMS Device Manager. You must modify this file to reflect your device.
***DEVICE*.MRG (Required)**	*DEVICE*.MRG is the database merge file you will create that contains your device template. (You create a device template as part of the process of building a Device Installation Kit.) When you create the merge file, you will need to change the default name of the file, as explained later in this chapter.
Alert File (Required)	This auto-generated file enables functionality in Alert Monitor, alert simplification, categorization, and consistency across devices supported in AMS Device Manager.
Alarm File (Required)	This auto-generated file enables alert functionality in DeltaV.
FHX File (Required)	This auto-generated file enables alert functionality in DeltaV.

4.2 Applications you will use

You will use the following applications to add your device to AMS Device Manager:

- Tokenizer - To generate the binary output file for your DEVICE.DDL file for adding the device to AMS Device Manager.

  Note

  AMS Device Manager version 10.5 adds support for tokenizer revision 8.x.

- AMS Device Manager - To create a device template and export that template to a merge file.
- AMS Device Manager Add Device Type utility - To test your installation kit files.
- AMS Device Manager Network Configuration utility - To configure the hardware you use to connect your device to AMS Device Manager (a HART modem or other interface).
- Map File Maker --automates the process of map file creation for a particular version of AMS Device Manager.
- Map File Tester -- used to test the file you created with Map File Maker

4.3 Assumptions

This chapter assumes that:

- You have installed AMS Device Manager.

  If you need more information about installing AMS Device Manager or selecting a data source, see the *AMS Device Manager Installation Guide*.

- You have a tokenized DD file.

4.4 Overview of steps

The following is an overview of the steps required for creating the installation kit. It is assumed that you have already created a working DD using the DD-IDE.

1. Make a directory for your new device and name it anything you like (for example, Device DD).
2. Run your source files through the tokenizer to generate .FMx (where FMx could be FMS, FM5, FM6, or FM8) and .SYM files.
3. Move the files generated in step 3 to the Device DD directory.
4. Copy the path of your Device DD directory.
5. Launch the Add Device Type utility of AMS Device Manager.
6. Click Browse and paste the address in the Path text box.
7. Click OK.

8. Click OK to run the utility.
9. Go to the installation directory of AMS Device Manager and open the AMS temp folder(c:\ams\temp).
10. Copy the generated files from your tokenized DD to your Device DD directory that you created in step 3.

    Ensure you do not overwrite the original contents of your Device DD directory like the .fm8 and its .sym file

11. Review the contents of the device ddinstal.ini file.

    Ensure applicable fields are filled out with the appropriate file names, manufacturer name, device type and descriptions. Also ensure the supported version includes the AMS Device Manager version you are using.

12. Create a template/mrg file.
    a. Add your DD to AMS Device Manager. Place the .fm8 in a folder by itself.
    b. Start Add Device Type, and browse to the folder where your .fm8 file is located, and click OK.
    c. Enter your company name, and the name of your device as you want it displayed in AMS Device Manager, and click OK.
    d. When Add Device Type creates your device, connect it to AMS Device Manager.
    e. Scan the device by right-clicking it and selecting Scan.

       This puts all the existing parameters in the AMS Device Manager database.

    f. Select View > Device Templates, and drag the device into the window.

       This creates a template.

    g. Open the template you created, and verify the values are correctly populated.
    h. Open Device Explorer and right click AMS Device Manager, and selectExport > To AMS Device Manager Export File...
    i. Uncheck all boxes, and check Device Templates.
    j. Press the Select button next to Device Templates, and uncheck all. Then find your device and check it. Press OK.
    k. In the Include Data section, enter a date range of a few weeks. Press OK.
    l. Navigate to \AMS\bin, and find the file <date>.ams_merge.
    m. Rename the file to <name>.mrg, and include the file in your device installation kit.

13. At this point, you have created a basic Device Installation Kit. The steps that follow allow you to take advantage of advanced features in AMS Device Manager 13.0 or to satisfy requirements of older versions of AMS Device Manager. The steps will vary depending on the version and features in the DD.
    a. If it is an enhanced EDDL device, it is supported by AMS Device Manager 7.6 and higher.

b. If you created a .MAP file, add it to the Device DD directory. Update the DDINSTAL.INI and retest the Device Installation Kit in AMS Device Manager.
c. If you have an alert (or .alm and .fhx file), add it to the Device DD directory.
d. If you have a .CHM file, add it to the Device DD directory. Update the DDINSTAL.INI and retest the Device Installation Kit in AMS Device Manager.
e. If you have a .PDF file for your device, add it to the Device DD directory. Update the DDINSTAL.INI and retest the Device Installation Kit in AMS Device Manager.
f. Submit your tested and verified Device Installation Kit to the AMS Device Manager Development Team.

4.5 Updating the DDINSTAL.INI file

1. Under the [General] heading, there is a line for NumOfMfg. This is the number of manufacturers on the diskette. If you are the only manufacturer who will be using this Device Installation Kit, the line should be:

   `NumOfMfg=1`

2. There is a heading for each Manufacturer, called [Mfgn] where n is the number of the manufacturer. If you are the only manufacturer using this install kit, the heading should be:

   `[Mfg1]`

   There are three lines in the [Mfg1] section; Name, ID and NumOfDev. Name represents the manufacturer name (for example, Fisher Controls); ID represents the 6-digit hexadecimal manufacturer ID (for example, 000013); NumOfDev represents the number of devices for this manufacturer on this Device Installation Kit (normally, this will only be one).

   Using these examples, your [Mfg1] section would look like this:

   ```
   [Mfg1]
   Name=Rosemount
   ID=000026
   NumOfDev=1
   ```

   The entry beside “Name” will be the manufacturer name displayed during Add Device Type.

   The entry beside “ID” is your hexadecimal manufacturer ID as assigned by the FieldComm Group.

3. The heading after [Mfg1] represents the first device on the Device Installation Kit by that manufacturer. The heading is represented by the 6-digit hex ID of the manufacturer, a dot, and a 1 (for the first device). For example, the heading might be [**000026.1**].

   Make sure this heading represents your manufacturer ID.

4. The first line under the ID heading is DevType. This is the 4-digit hex ID for the device type. For example, if the device type ID is 0006 in hex, the line would be:

`DevType=`**`0006`**

5. The second line is the description of the device type. For example, for a 3051C, the line would be:

   `DevTypeDesc=`**`3051C`**

6. The third line is a text description of the device revision. For example, for a device revision of 3, this line would be:

   `DevTypeRevDesc=Rev `**`3`**

7. The fourth line is the actual revision of the device (2 digits). For example, for a device revision of 3, this line would be:

   `DevTypeRev=`**`03`**

8. The fifth line is a text description of the DD revision of the device. For a DD revision of 2, this line would be:

   `DDRevDesc=DD Rev `**`2`**

9. The sixth line is the name of the .FMx file for the installation kit. For example, if the installation kit has the name of the .FM8 file is 0302.FM8, this line would be:

   `FMS=`**`0302`**`.FM8`

10. The seventh line is similar to the sixth, except that it is the name of the .SYM file. For example, if the .SYM file is 0302.SYM, this line would be:

    `SYM=`**`0302`**`.SYM`

11. The eighth line is similar to the previous two, except that it is the name of the .MAP file. For example, if the .MAP file is 0302.MAP, this line would be:

    `MAP=`**`0302`**`.MAP (if no map file, leave blank)`

12. The next few lines are the names of any general device help file, block help file, or other type of device help file (including pdf documentation). If you do not have any help files, use "xxxx.chm" or "xxxx.pdf". Otherwise use your file names. For a device with no additional help files an example of these lines would be:

    ```
    DeviceHelp=xxxx.chm
    BlockHelp=xxxx.chm
    OtherHelp=xxxx.chm
    OtherHelp=xxxx.pdf
    ```

13. The next line is for your device DLL file. Replace the bold text with your device name. The line would be typically be blank:

    `DeviceDLL=`

14. The next line indicates whether you have any other DLL files. If you have no other DLL files, the line would be:

    `OtherDll=`

15. The next line is the name and diskette location of the merge file. For example, if the merge file is named 3051C.MRG, the line would be:

    `Template=`**`3051C`**`.mrg`

16. The next line indicates the HART protocol revision of the device. If the device is HART revision 5, the line would be:

    ```
    HART Protocol Rev=5
    ```

17. The final line indicates which AMS Device Manager versions your device operates with. For example, if your device is supported by AMS Device Manager versions 10.0-11.0, the line would be:

    ```
    AmsVersionSupport=10.0,10.5,11.0
    ```

18. Save your changes and close the DDINSTAL.INI file.

4.6 Troubleshooting

If you have any trouble with your Device Installation Kit, the following troubleshooting suggestions may be helpful:

- If the device for which the new template is being created is displayed as a generic device by AMS Device Manager, check the following:
    - Have the device files been added to the correct place in the directory structure?
    - Has the FMS.INI file been properly updated?
- If the merge file contains more than one manufacturer, device revision, or device template, more templates than were required were selected as items to be exported.
- If AMS Device Manager does not show the live device on the modem, do the following:
    - Use the AMS Device Manager Configuration program to ensure that the modem has been assigned to a COM port.
    - Check that the modem is connected to the serial port corresponding to the COM port.
- If AMS Device Manager issues error messages when it tries to populate the database with the new device or when making the template for the new device, do the following:
    - Run the AMS Device Manager Database Verify/Repair operation (select Start > Programs > AMS Device Manager > Database Utilities > Database Verify Repair from the Windows taskbar).
- If the event section of the merge file is empty, the time range specified for the export did not include the correct times and dates. Try exporting again, making sure that the date and time when you made the template is included.

5 Installing and testing your device files

This chapter explains how to install your device into an existing AMS Device Manager system to test your device interface.

5.1 Hardware you will use

You will need the following hardware to test your device:

- A HART modem or other supported HART interface - To communicate between your device and AMS Device Manager.
- Your device, connected to AMS Device Manager.
- A power supply for your device.
- Load resistor, typically 250 ohms (consult the device manual).

5.2 Assumptions

This chapter assumes that:

- You have installed AMS Device Manager.

  If you need more information about installing AMS Device Manager or selecting a data source, see the AMS Device Manager Installation Guide.
- You have created a Device Installation Kit.

5.3 Overview of device installation and test steps

5.3.1 Step T-1: Configure your AMS Device Manager network and connect your device

After you have created your Device Installation Kit, you are ready to connect your device to AMS Device Manager in order to test your device.

Refer to the *AMS Device Manager Installation Guide* for information on how to connect your device to AMS Device Manager.

Note

For information on simulating your device in AMS Device Manager, see *Appendix B*.

5.3.2 Step T-2: Start AMS Device Manager and test your device files

Once your device is connected to AMS Device Manager, test the functionality of your device:

1. Start AMS Device Manager (refer to the AMS Device Manager Books Online if you need instructions).
2. Open the Device Connection View, if it is not already open (View > Device Connection View). You should see your device attached to the modem.

Note

The first time you run AMS Device Manager with your device connected, there will be a short delay as AMS Device Manager initializes its database with your device information.

3. Scan your device and verify correct operation/completion.
4. Double-click your device icon to open the faceplate and verify that it looks as intended.
5. Right-click on the faceplate to open the context menu. Verify that the context menu looks as intended. Verify the operation of all context menu items.
6. Verify all of the methods available from the device screen as needed and test each method, ensuring that they complete without errors.
7. Verify that the Status/Device Diagnostics screen appears as intended. Launch any other context menu items.
8. Verify that the Configuration Properties screen looks as intended. Verify that each parameter intended to be editable from the Configuration Properties screen is editable. Ensure that all writable values on the screen actually write to the device successfully when you select Apply. Verify labels, tab order, and parameter Help.
    a. Read each configuration parameter (current, history).
    b. Verify that the data appears to be valid (for example, no * are in any fields).
    c. Write each configuration parameter (current).
    d. Reopen and verify the newly written parameters.
    e. Verify PRE_EDIT_ACTIONS, POST_EDIT_ACTIONS, PRE_READ_ACTIONS, POST_READ_ACTIONS or PRE_WRITE_ACTIONS, if applicable.
9. Verify refresh relations and unit relations, if applicable.
10. Ensure that all writable values on the screen actually write to the device successfully when you select Apply.
11. Verify that the Compare Configuration screen looks and functions as intended. Verify that each parameter intended to be editable from the Compare screen is editable. Verify labels, tab order, and parameter Help. Perform the following steps for the device:
    a. Read each current, history configuration data for each parameter (source and destination).
    b. Verify that the data appears to be valid (for example, no * are in any fields).

  c. Transfer config data.
  d. Write any current configuration parameter (source and destination).
  e. Reopen and verify the newly written parameters.
  f. Verify PRE_EDIT_ACTIONS, POST_EDIT_ACTIONS, PRE_READ_ACTIONS, POST_READ_ACTIONS or PRE_WRITE_ACTIONS, if applicable.

12. Verify refresh relations and unit relations, if applicable.
13. Verify that an offline configuration can be created and edited correctly.
14. Verify that all Help strings are displayed as intended.
15. For each mode, if applicable, verify that all displays look and function as intended.
16. For each mappable dynamic variable, if applicable, verify that all displays look and function as intended.
17. Execute Diagnostic and Test methods multiple times using multiple paths. Verify that the text on the dialogs displays completely and that the method completes without error.
18. Execute Calibration Methods multiple times using multiple paths. Verify that the text on the dialogs displays completely and that the method completes without error.
19. Execute a Master Reset and verify the proper dialog display and test completion.
20. Verify that Process Variable parameters can be read, updated, and will launch other context items.
21. Verify that you can change the name of the device.
22. Verify that the drawings/notes for the device can be entered and saved correctly.
23. Verify that entries are being created correctly in the Audit Trail. Select an individual entry and verify that multiple values are defined as having been changed.
24. Verify the ability to switch between modes of Multi-Mode Devices.
25. Set up a route in AMS Device Manager and as you follow the steps to input calibration data, ensure the range and units listed match your map file.

Note

These steps for testing are not meant to be all-inclusive, but only to be a guideline for testing your device files.

5.3.3 Step T-3: Modify your Device Installation Kit

If you discover a problem with your device in Step T-2, edit the file to fix the problem and rebuild your Device Installation Kit.

You can also restore the AMS Device Manager database from the backup you performed in Section 2.

5.3.4 Step T-4: Copy your new file into the AMS directory

Once you have modified the file and rebuilt your Device Installation Kit, test your changes by copying the file into the Device Installation Kit you created and rerun Add Device Type.

5.3.5 Step T-5: Repeat steps T-2 through T-4 as needed

While you are testing your device, you may decide to make some changes. If so, repeat Steps T-2 through T-4 until you are satisfied with the functioning of your device with AMS Device Manager.

As part of this testing, you may need to modify the DD for your device. If so, make the modifications, tokenize the DD, update your install kit, and reinstall.

When you complete your testing, restore the AMS database and ensure your device is working with a clean AMS database.

The following paragraphs contain some additional, specific testing to perform on your devices, if applicable.

Simulating your device using AMS Device Manager

AMS Device Manager can simulate the basic functions of your device so that you can test the interface without having to physically connect your device to AMS Device Manager. To set up AMS Device Manager to simulate a device, please refer to *Appendix B*.

Log files for debugging your device

Look in the application log of the event viewer for AMS Device Manager errors.

The HARTLog captures and displays HART packets to and from AMS Device Manager and your device. The HARTLog can be enabled by manually invoking the AMS Device Manager server component with the -l (logging) option, **prior to starting AMS Device Manager**, and giving it the path and name to the resulting log file. To do this, edit the following entry in the FMS.INI file (in your Windows folder):

```
[AMS Server]
CmdLine=-l c:\temp\hartlog.txt
```

You can substitute a filename for the one above.

The HARTLog file grows quickly, so you should clear this setting in the FMS.INI file whenever it is not needed.

You can also review the ddinstal.log file, located in the \AMS\Log folder. Look for "ERROR" near the end of the file.

When testing your device, you should use these files to ensure that AMS Device Manager is not generating error messages that are not being reported to the user interface.

Clean database

Make a backup of your AMS Device Manager database before you add your device. Later you can restore the database without the device, if necessary.

Troubleshooting

If you have any problems testing your device files, you may find the following troubleshooting suggestions helpful:

- If the device being tested is displayed as a generic device by AMS Device Manager, check the following:
  - Have the .FM6 or .FM8, and .SYM files been added to the correct place in the directory structure?
- If AMS Device Manager does not show the live device on the modem, do the following:
  - Use the AMS Device Manager Configuration program to ensure that the modem has been assigned to the correct COM port.
  - Check that the modem is connected to the correct serial port.
- If AMS Device Manager issues error messages when it tries to populate the database with the new device, do the following:
  - Run the AMS Device Manager Database Verify/Repair operation (select Start > Programs > AMS Device Manager > Database Utilities > Database Verify Repair from the Windows taskbar). See the AMS Device Manager Books Online for more information.

6 Integrating the device into AMS Device Manager

This chapter explains how to integrate your device into AMS Device Manager once the installation kit has been created.

6.1 Methods for integrating into AMS Device Manager

There are two methods for integrating the device into AMS Device Manager. The first is through an independent integration which calls for distribution by your company of the installation kit for the device created in the previous chapter and sending this kit directly to the customer. The second method is through a full integration, whereby the device is distributed with AMS Device Manager to all customers automatically.

Your device should be registered with the FieldComm Group prior to distribution.

Independent integration into AMS Device Manager

Using the Device Installation Kit created in the previous chapter, distribute this kit to customers requiring the device operation in AMS Device Manager.

This method is also useful when putting out modifications or new releases of devices that do not coincide with a scheduled release of AMS Device Manager.

Additionally, Emerson has developed a web site (*http://www2.emersonprocess.com/en-US/documentation/deviceinstallkits/Pages/deviceinstallkitsearch.aspx*) to provide a convenient location to download Device Installation Kits, and for device developers to make updates available more quickly to their customers.

You can email questions about this site to: *amsdevicemanagertoolkit@emersonprocess.com*.

Full integration into AMS Device Manager

This method allows the device to be included with the release of AMS Device Manager to all customers. The AMS Device Manager build process requires all the files that are normally distributed with the installation kit.

Device Description .FM6/ FM8 and .SYM	This represents all source files required to tokenize the DD to the .FM8 or .FM6, and .SYM files, and any other device-specific files imported into the DD. It is critical that your device .SYM file is synchronized with the HCF. Use the SYMBOLS.TXT file from the FieldComm Group to tokenize your DD or use the DD you registered.

***DEVICE*.MAP (Optional)**	This file contains mappings from device-specific parameters to a common format used by calibrators and export/reporting applications.
Device Merge File	The device merge file, created in the previous chapter, contains a default template for the device as well as information about the manufacturer and device.
DDINSTAL.INI	This file shows AMS Device Manager how to install the device. It is incorporated into the overall install file with all other integrated devices.
Alert File (Required)	This auto-generated file enables functionality in Alert Monitor, alert simplification, categorization, and consistency across devices supported in AMS Device Manager.
Alarm File (Required)	This auto-generated file enables alert functionality in DeltaV.
FHX File (Required)	This auto-generated file enables alert functionality in DeltaV.

6.2 Integration timeline

Device Installation Kits can be submitted at any time to *amsdevicemanagertoolkit@emersonprocess.com*. They will be integrated into the next release of AMS Device Manager and placed on the AMS Device Manager DD download website.

6.3 The next step

If you have questions about the integration process or any issues with the integration of your device into AMS Device Manager, send an email to: *amsdevicemanagertoolkit@emersonprocess.com*.

Appendix A

AMS Device Manager Device context-sensitive Help - getting help beyond DD help strings

This appendix contains a description of the procedure for creating context-sensitive "Advanced" device help. Also included is information about providing a product manual in .PDF format.

A.1 Introduction

This chapter shows device developers how to create device-based online help that extends beyond textual help provided in the DD. As of AMS Device Manager 10.0, this help may consist of:

- Advanced Help for a device parameter is displayed by clicking the More button within a DD help window. See *Section A.2*.
- A Product Manual, provided in .PDF format. See *Section A.4* for an example.

In addition to this appendix, there are sample files provided to guide you through creating Advanced Help, and providing a .PDF file. See the folder User Assistance for sample files you may want to use in creating your system.

A.2 Providing Advanced Help

Advanced Help for EDD devices involves information beyond the standard EDD Help string. Beginning in version 8.x, you can provide Advanced Help by connecting parameter names of VARIABLE, GRAPH, CHART, and IMAGE to topics in your Help file.

Advanced Help for devices is in context to the DD Help window that a user launches, and can be done in HTMLHelp.

Assumptions

AMS Device Manager requires that a Help attribute be defined in the EDD for every parameter for which you provide Advanced Help. A user links to Advanced Help from the EDD Help window.

You can provide one .CHM file per Device revision. HTMLHelp is a set of HTML topics compiled into a binary format(.CHM). These compiled files are viewable on any Windows PC with Internet Explorer. The end user does not need to install or associate any applications to view the file.

To create a CHM file for your device, you must have Microsoft HTMLHelp Workshop. Microsoft HTMLHelp Workshop is the free Windows Help compiler available from Microsoft at *http://www.microsoft.com/download/en/details.aspx?id=21138*.

Although you may find it helpful to purchase a third-party tool for creating HTMLHelp, it can be done with an HTML editor and Microsoft HTMLHelp Workshop alone. A very inexpensive tool that handles .HHP and .HHC editing better than the compiler is called FAR. It can be purchased from www.helpware.net.

The source files for an HTMLHelp system shall consist of the following:

- The project file (.HHP) – The project file contains the settings for the Help system.
- HTML file(s) – The content.

The .<device.ini> file – The file that maps the parameter names to the HelpIDs. The files you deliver as part of your DDinstal files consist of the following:

- The .CHM file – The compiled HTMLHelp file. It must be where the application expects it.

A.2.1 Create an HTMLHelp system for your device

Before you start creating your Help system, you must decide on a name for it. This is the name that will be used for source files and the files that are delivered to the user. A good nomenclature is <devicetype><devicerev><ddrev>.chm.

1. Determine which EDD parameters will need Advanced Help.
2. Create the [ADVANCED HELP] section of the <device.ini> file.

   ```
   [AdvancedHelp]
   buffer_standard=16894
   device_status=151
   glass_high_fault=16901
   glass_low_fault=16902
   hold_config=16873
   ```
3. Create the map file.

   Copy the values defined in the ADVANCED HELP section of the <device>.ini to a <device>.h file and reference it in your project (see [MAP] section of sample .HHP file).

   The HTML Help compiler expects a format nearly identical to the [ADVANCED HELP] section. Simply replace the = sign in with a space, and your IDs will be mapped.
4. Create the project file.

   The HTML Help compiler expects a format nearly identical to the [ADVANCED HELP] section. Simply replace the = sign in with a space, and your IDs will be mapped.

   ```
   [MAP]
   #include <device>.h
   ```

If you compile the project with the #include statement above, the device.h file must be in the same folder as the .HHP. Create an [ALIAS] section, and map the Topic ID to the .HTM or .HTML file with the following format:

```
<DD parameter name>=<path to .htm file>;<comment>
HID_DEVICE_WINDOW_CONFIGURE_VIEW=DEVCONF_OVER_ENHANCED_DEVI
CE_WINDOW.htm ; Configure view
```

This section assumes the .HTM or .HTML files are in the same folder as the .HHP.

For more information on the project file settings, see the Microsoft HTMLHelp Workshop online help.

5. Create the topics.

   Create a folder for your project, and then launch your HTML editor and create the HTML file(s) that will contain your help topics.

   Your files must be valid HTML 4.0 or valid XHTML.

   You will be creating one topic per device parameter.

   See the online help in Microsoft HTMLHelp Workshop or the following MSDN articles for more information on creating topics:

   http://msdn.microsoft.com/en-us/library/ms670169

   http://msdn.microsoft.com/en-us/library/ms669980.aspx

6. Create the table of contents.

   To create a table of contents that appears on the Contents tab in the Help viewer, you can use the HTMLHelp Workshop HHC editor. There are also 3rd party tools available for this.

7. Create the index.

   To create an index that appears on the Index tab in the Help viewer, you can use the HTMLHelp Workshop HHK editor. There are also 3rd party tools available for this. By default, keywords are maintained in a separate file, but you can also use HTML <META> tags to keep the keywords in the file they describe. When editing the project file (.HHP), select the Change Project Options button on the top left, and select the option “include keywords from HTML files”.

   Follow these guidelines when you create the index:

   - Check the index of AMS.chm to determine if your index term is being used consistent with that document’s index.
   - Index to no more than two levels.
   - Avoid single secondary keywords.
   - Only capitalize the first word of an index entry if it is a proper name.

8. Compile the project.

To create the .CHM file, you compile your Help system. To compile your help system, click in Microsoft HTMLHelp Workshop. If you have set the project file to display notes, you will see a list of messages about the compilation. Look at the messages to see if they indicate errors that you need to address.

9. Test the project.

   After successfully compiling the file, copy it to the proper device folder and then test it in AMS Device Manager.

   a. Modify your Device Installation Kit.
   b. Copy the .CHM to the installation kit folder.
   c. Update the DD install.
   d. Device help = YourHelp.CHM
   e. Run the Add Device Type utility to add the files to the system.

   You are responsible for testing the content and integrity of your help system. A unit test plan included in the sample will give you some ideas about how to test a compiled HTML Help file. It is not an exhaustive list.

   Note
   AMS Device Manager is **NOT** responsible for testing the content of your help file.

   The parameters with help should display the DD help with a “More” button. Clicking this button will bring up your help file in context.

10. Add the Help files to the install kit.

    When the Help for the device is complete, add it to your device file shipment by copying the .CHM into the folder that contains the rest of the device files. When we know you are providing Advanced Help, we will modify AMS Device Manager to enable launching of the files. Make sure that your files are included in your device integration checkout process or process approval form.

A.3 Writing guidelines

Conventions

The following writing conventions should be used to ensure consistency with existing product online Help.

Design-test your device’s UI with multiple users outside your development organization. A good UI can alleviate the need for extensive Advanced Help.

Content

Follow these guidelines when writing the content for context-sensitive (More button) help:

- The user’s primary help for your device should be a .PDF or DD-based help. Making the user click a More button should only be done if the explanation for the parameter in DD help is insufficient.
- The potential values/entries should be explained in the context of the mode of the device so the user can be guided towards making the right configuration settings. Explain what fields are required to be addressed by the user and which ones are not.
- Provide troubleshooting sequences, recommended actions for alerts, and practical advice for the technician/engineer when appropriate.
- Where appropriate, use graphics, and links to other topics, See the Sample Help System provided with this toolkit for instructions on how to include these in your help topics.

Writing style and usage

Follow these styles and rules when writing your content:

- Use active voice.
- Use present tense wherever possible. Future tense is appropriate occasionally, but in most cases it is not.
- Use second person ("you") rather than the third person ("the user") wherever possible.
- Be as concise as possible, using only as many words as needed to get the meaning across. For example, use: "lets you" instead of "allows you to"; "To..." instead of "If you want to..."
- Use an informal—but not chatty—writing tone.
- Do not use contractions.
- Use “click” not “click on” to designate selecting a button using the mouse (“Click Help”) except for something like “by clicking once on its label”.
- Use login as a noun form “the login screen” vs. log in as a verb “log in to AMS Device Manager”. Do not hyphenate.
- Use "on a tab" vs. "in a tab."
- Use "want" vs. "wish."
- Use "PC" or "computer" vs. "machine."
- Use "window" vs. "screen" when referring to a view in an application.
- Use "dialog" when referring to the type of window that elicits a response or input from a user.
- Lists:
  - Introduce a list with a complete sentence. This aids in translation efforts.
  - Use unordered (bulleted) lists for lists of items.
  - Use numbered lists for first-level procedures and lettered lists for second-level procedures.
  - Make items in a list parallel.
  - Capitalize the first word of each item in a list.

  - End each entry with a period if all entries are complete sentences, are a mixture of fragments and sentences, or complete the introductory sentence. An exception is when all entries are short imperative sentences (three words or fewer) or single words; these entries do not need a period. If all entries are fragments that do not complete the introduction, do not end them with periods.
  - Use hyphens (-) vs. em dashes (—) to explain items in a list (for example, "AMS Tag - The AMS tag for the device").
- Notes:
  - Use notes sparingly. Use only for exception or aside information. Do not overuse notes or they lose their value.
  - In HTML Help topics, do not use multiple notes under one “Notes” heading. Multiple notes must be implemented as separate back-to-back notes. Avoid back-to-back notes if at all possible. If back-to-back notes are absolutely necessary, avoid using more than two.
- Titles:
  - Use the imperative mood for the verb in a procedure topic title in HTML Help ("View the Audit Trail for an AMS tag", "Scan devices").
  - When an overview topic is about an action, use the gerund form of the verb in the title ("Importing and exporting data to and from the ValveLink standalone application", "Troubleshooting Drawings/Notes").
  - Use "Subject " instead of "About subject" in an overview topic title, except in titles that use gerunds. Never use the word ‘Overview’ in a title with a gerund (for example “Replacing Assigned devices”). Avoid using “Subject Overview” unless there is enough information about a subject to warrant multiple reference topics. In that case the Overview topic is that subject’s “home page” and the destination for links. In titles with gerunds, do not use the word "overview" (for example, "Replacing Assigned devices").
- When the text refers specifically to words or numbers that appear on a display, the text should exactly duplicate the capitalization and punctuation as it appears on the display. For example: “In the Configuration window, select Offline in the Time field.” In this example the word "Offline" is capitalized because it is capitalized on the display. But in the following example, the word "offline" is not capitalized, because it is used in a generic sense: “Only one offline configuration can be associated ...”
- Use "offline" and "online" as single words, not hyphenated.
- Use startup as a noun (“faster startup”) and start up as a verb (“start up User Manager”). Avoid the ambiguous hyphenated form “start-up”.
- Indicate use of a key on the keyboard by the word “press” (for example: “... press ENTER to commit the changes”). Do not say “... press the ENTER key ...” or “... press down the ENTER key.”
- Numbers:
  - Common practice suggests numbers between zero and nine be written out, presumably to increase their readability. Invariably, the best rule to follow is “watch the context”.
  - Be consistent when dealing with multiple sets of numbers in a single sentence.
  - When describing the User Interface, use what the UI uses.

  - Technical documentation, with its focus on clarity, often relies on numerals for its precision. Numerals should almost always be used when defining ranges, indicating date or time, capacity (with units), or default settings.

Examples:

This option lets you select four types of database problems.

This value must be an integer in the range 1 through 14.

A minimum of 5 test points for one direction or 9 points for two directions are required to calculate zero, span, or linearity.

Writing for translation

There are many resources, publications, and associations available to assist in writing for translation. The Globalization and Localization Association (GALA) is one well-known association. Improving translatability in the English document avoids repetitive costs translating and fixing problems for each language. This list is a small portion of issues that frequently arise. Consult your translation or localization vendor prior to beginning an effort for further ideas.

Writing tips

- Avoid using the construction “(s)” to indicate “either singularity or plurality”. This usually translates into poorly-constructed non-English sentences, and adds cost to translation. Whenever possible, choose either the singular or plural. (ex. Poor: “Enter the point(s) to complete the transaction.” Better: “Enter any points to complete the transaction.”)
- Strive to introduce bulleted lists with complete sentences.
- Develop a structured language with terms that are allowed. Avoid using synonyms whenever possible to improve consistency.
- Avoid ambiguity, such as the following: “Once charging ends, unplug the unit.” In this context “once” could mean “when”, “after” or “if”.
- When describing amounts or units, use the International System of Units (SI). It is a standardized list of acceptable abbreviations. If possible, and when supported by your text editor, use text entities for units rather than combinations of letters.
- Avoid unnecessary and difficult words. (For example, use “total”, not “aggregate”)
- Be consistent and reuse blocks of text.
- Globalize references to date, time, address, and phone numbers.
- Avoid subordinate clauses.
- Avoid nouns created from verbs. (For example, “refuse” vs. ”refusal”)
- Match names of menus, buttons, and other UI labels exactly (or as closely as possible) in the documentation.

Layout and design tips

- Separate graphics from text.
- Avoid changes to the format or text for stylistic reasons.

- Clean up formatting (such as extra carriage returns) and use styles.
- Provide white space for tables and text to expand. Translated text will expand by about 30 percent.
- Be aware of color theme preferences by region.
- Use common fonts.

Abbreviations and acronyms

- Define acronyms in the glossary.
- In HTML Help topics, do not spell out the first occurrence of an abbreviation or acronym in a topic. Instead, link the first occurrence to its definition in the glossary, but only if the abbreviation or acronym is one that may not be familiar to some users. For example, do not link "HART" to its definition in the glossary, because HART is a well-known term.

  In user manuals, show the full spelling of an abbreviation or acronym the first time it is used (put the spelled-out form first, followed by the abbreviation in parentheses).
- Do not use Latin abbreviations: do not use i.e., e.g., or etc.; instead, spell out the words “that is,” “for example,” “and so forth.”
- Do not use "via." Use "by," "through," or "by means of" instead.
- Spell out any other abbreviations that might be misunderstood by the reader.
- Spell out symbols such as “percent” and “dollars” rather than using % and $ in text. However, use symbols where there is a space limitation, as in tables or figures.
- Form the plural of an abbreviation by the addition of a lowercase s. Do not use ’s, unless the meaning might be confused or the abbreviation contains periods.
- Use “FF" as the abbreviation for FOUNDATION fieldbus only when “FF” is used in the user interface, and even those occurrences should be used sparingly.

Capitalization

- Capitalize proper names. Proper names include names of objects in the user interface, such as icons, windows, dialog boxes, fields, and buttons; and pieces of functionality (such as Alert Monitor or Audit Trail). Leave the word for the object (“window”, "dialog box") in lowercase (for example, "Compare Configurations window", "Export dialog box").

  The decisions about what are proper names are reflected in the glossary for the product. (The decisions may sometimes seem arbitrary.) If you are unsure about whether an item is a proper name, you should consult with other members of the writing team.
- Capitalization of an item in the UI does not necessarily mean the item is a proper name.
- Terms used in their “verb form” are not capitalized. (For example, "The alert monitoring...".)
- Capitalize only the first word and proper names in headings (all levels)
- Capitalize only the first word and proper names in HTML Help topic titles (the <title> attribute) and in Related Topics links.

- In user manuals, capitalize the first letter of significant words in figure and table titles, section titles, and headings.
- When referring to online Help, the word "Help" is capitalized.
- Full-cap HART (as in HART protocol or HART device). HART is an acronym for Highway Addressable Remote Transducer.
- Full-cap and hyphenate SNAP-ON. Always follow the term SNAP-ON by the term “application,” for legal reasons.
- When referencing another section of a user manual, initial-cap the word "Section".
- Use "FOUNDATION fieldbus" in high-level topics and "fieldbus" everywhere else. Use "fieldbus" and "FOUNDATION fieldbus" interchangeably, with the following standardization:
    - When the usage is "FOUNDATION fieldbus," or when the word "fieldbus" is used alone, it is never capitalized.
    - "FOUNDATION" is initial capitalized AND written in all small caps when the usage is "FOUNDATION fieldbus."
- Avoid the abbreviation FF in most cases. Spell out FF as “FOUNDATION fieldbus" (use all caps if small caps is not available in your font).

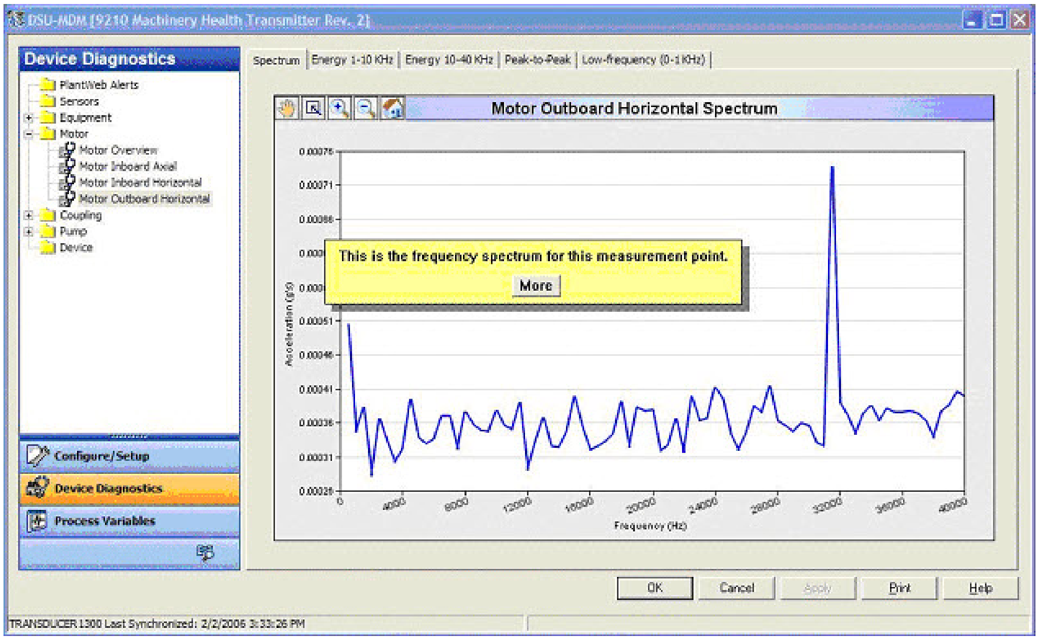
Punctuation

- Use one space after a period.
- Use hyphens, en dashes, and em dashes appropriately (follow the model already in use).
- When a singular noun ends in ‘ss’, and it needs possession, ss’ is preferred over ss’s (example boss’, business’).
- Use commas to separate ALL items in a series (for example, “device configurations, user configurations, and device templates”).

Example of More button

1. User clicks the What’s This? button in AMS Device Manager toolbar and then clicks on a parameter.
2. If Advanced Help is available, the More button is displayed as shown in *Figure A-1*.
3. When the user clicks the More button, the associated Advanced Help is displayed.

Figure A-1: Example of a More button



DSU-MDM [9210 Machinery Health Transmitter Rev. 2]

This is the frequency spectrum for this measurement point.

TRANSDUCER 1300 Last Synchronized: 2/2/2006 3:33:26 PM

A.4 Providing a product manual

You should include a product manual in PDF format with the AMS Device Manager shipment. The product manual is accessed from two locations in AMS Device Manager:

- From the Help > Device submenu, where a user selects your device from a list.
- From the Help selection on your device’s context menu.

You may send a single product manual for each DD revision you are supporting in AMS Device Manager, or you may ship one manual that opens for each device. If you choose not to provide a product manual, your device will not appear in the Help > Device submenu, and a generic topic from AMS Device Manager will display when the user chooses the Help selection on the device context menu.

Your product manual should follow the guidelines listed in *Section A.3*, and should clearly indicate the device, device revision, and DD revision (if applicable). Emerson devices should follow appropriate branding guidelines. The name for your Product Manual in AMS Device Manager should indicate the device(s) to which it refers. A good nomenclature is <devicetype><devicerev><ddrev>.PDF.

Add the PDF to your device files folder

Add your .PDF to your device file shipment by copying the .PDF file into the folder that contains the rest of the device files. When we know you are providing a product manual, we will modify AMS Device Manager to enable launching of the files. Make sure that your files are included in your device integration checkout process or process approval form.

Test

Modify your device's DDINSTAL.INI file to Device_Help=<yourfilename.PDF>. This allows AMS Device Manager to launch the PDF from the Help > Device submenu, as well as the Help selection on your device's context menu.

Note

AMS Device Manager is **NOT** responsible for testing the content of your product manual file.

Appendix B
Simulating devices in AMS Device Manager

This appendix serves to describe the procedure for generating simulation files for AMS Device Manager.

Simulation files are useful for exploring the look and feel and functionality of a device without having to connect to an actual live field device.

Simulation files do have some known issues. These shortcomings are covered later in this appendix.

B.1 Introduction

The use of this appendix is intended to serve three purposes:

1. Provide a step-by-step method of creating simulation files (or sets of simulation files)
2. Describe the procedure for changing values in a simulation file
3. Describe shortcomings and other items to be aware of while using simulation files

The following software tools are assumed to be in use:

- AMS Device Manager 13.0 or higher
- AMS Device Manager Network Configuration Utility
- Text Editor (Wordpad, Notepad, etc.)

As an example, we will use the Rosemount 3051 Revision 6 device (manufacturer 0x26, device type 0x06, device revision 0x06, DDrev 4).

B.2 Steps to simulate a device in AMS Device Manager

The following steps set up a device for simulation in AMS Device Manager.

B.2.1 Step 1: Generating the simulation file

A modem can connect to a maximum of 16 devices at a time (polling address 0 through 15). The simulation file consists of one record for each simulated device. Each record consists of the following information, in the following order (all number fields are in decimal):

- Polling address

- Manufacturer ID
- Device Type ID
- Device Revision
- DD Revision
- HART Protocol Revision
- Unique Simulation Device ID
- Comments (optional)

The file is space- or tab-delimited. Any number of spaces is acceptable. Therefore, the file can be formatted for readability or simply for functionality. Here is an example of a record for a few revisions of our sample device:

```
#Poll  Man  Dev  Rev DDRev HART ID   Comments
# 0    38   6    2   6       5  100  #(26060206) Rosemount 3051
r2
# 1    38   6    3   2       5  101  #(26060302) Rosemount 3051
r3
  2    38   6    6   4       5  102  #(26060602) Rosemount 3051
r6
```

Two of the three revisions are “commented out” by leading the line with the number sign. This means they will not be simulated.

Any number of devices can be included in this file; even multiple lines with the same manufacturer, device, and revision. However:

- A modem can only connect to 16 devices (polling address 0 through 15)
- A given polling address can only be used once
- Simulation device ID numbers must be unique

A standard text editor can be used to generate this file. The file must be named HrtsimXX.dat, where XX is the simulation ID number. This must match the number specified for the simulated modem number in the AMS Device Manager Network Configuration utility.

B.2.2 Step 2: Setting up AMS Device Manager

AMS Device Manager, by default, is not set up to run simulation. Also, whenever you add a modem to AMS Device Manager, it also is not set up for simulation by default.

1. Using a text editor, open up the FMS.INI file, located in your Windows folder.
2. Ensure that the following entry is set:

   ```
   [INSTALLATION]
   Simulator=Yes
   ```
3. Save the file.

If you do not have this setting correct, the simulation options will not be visible when you are adding a modem in the AMS Device Manager Network Configuration utility.

B.2.3 Step 3: Turning on simulation in the modem

Assuming you have already added the modem you wish to simulate, do the following:

1. Open the AMS Device Manager Network Configuration utility.
2. Display the properties for the modem you wish to simulate.
3. On either the Advanced or Connection tab (depending on the modem you are working with) there should be simulation settings.
4. Ensure that the simulation option is checked.
5. Ensure that the simulation file ID is set to match the number of the HrtsimXX.dat file you have set up.
6. If you are using a simulated modem and plan to simulate multiple devices at the same time, check to ensure that the polling addresses are set to a wide enough range to handle your devices (for example, 0 through 15).
7. Save the settings and exit the utility.

B.2.4 Step 4: Create the simulated field device files

The simulation set has been created. Now you need to have the AMS Device Manager Server create the actual field device simulation files. There will be one file created for every device being simulated. These files contain the actual parameters and their associated structure information and values. AMS Device Manager will mimic communication to a live field device by reading and writing to these files instead.

To accomplish this step, do the following:

1. Using a text editor, open the FMS.INI file, located in your Windows folder.
2. Ensure that the following entry is set:

   ```
   [AMS Server]
   Cmdline=-i X modem
   ```

   ...where X corresponds to the HrtsimXX.dat file you are using.
3. Save the file.
4. Right-click the AMS Device Manager Server icon in the system tray and select Start AMS Device Manager Server.
5. Wait until the AMS Device Manager Server shuts down (shown by a red line through the icon). This may take several minutes, depending on the number of devices being simulated.
6. Open the FMS.INI file, if it is not already.
7. Modify the following entry to read:

   ```
   [AMS Server]
   Cmdline=
   ```
8. Start AMS Device Manager. Your simulation should appear.

B.3 Modifying simulated device files

Each simulated device has its own simulation file, MMDDXXXX.INI, where MM represents the two-digit hexadecimal representation of the manufacturer ID, DD represents the 2-digit hexadecimal representation of the device type ID, and the XXXX represents an instance ID generated by AMS Device Manager. For example, our Rosemount 3051 Revision 6 would have a simulation file: 2606AC59.INI.

Structure of the device simulation file

Each parameter the device “knows” about is represented by a row in the device simulation file. The following fields are represented by each row, in order:

- symbol ID (decimal)
- parameter label (if available)
- variable type
- length (bytes)
- value

Here is an example of a portion of the file for our device:

```
150, "", 6, 1, 0
151, "", 7, 1, 0
152, "", 7, 1, 0
153, "Manufacturer", 6, 1, 38
154, "Model", 6, 1, 6
155, "Num req preams", 3, 1, 5
156, "Universal rev", 3, 1, 5
157, "Fld dev rev", 3, 1, 6
158, "Software rev", 3, 1, 1
159, "Hardware rev", 3, 1, 1
160, "Dev flags", 6, 1, 2
161, "Dev id", 3, 3, 1012
162, "Poll addr", 3, 1, 12
163, "Tag", 10, 8, 0C260606
164, "Message", 10, 32, MESSAGE_901234567890123456789012
165, "Descriptor", 10, 16, DESCRIPTION.....
6004, "Lo snsr trim pt", 4, 4, 3.402823e+038
6006, "Up snsr trim pt", 4, 4, 3.402823e+038
```

The value for symbols 6004 and 6006 are both 3.402823e+08. This arbitrary value is set when the AMS Device Manager Server creates the device simulation file. It only represents a default simulated value.

Changing the value field for a particular parameter is exactly like making a change to a parameter in a live field device (note shortcoming described below).

Alternative method of locating parameters

For parameters that have labels, you can easily find the AMS Device Manager-displayed parameter row in the simulation file. However, not all parameters have this label listed in the simulation file. For these values you can find the associated parameter name and its corresponding symbol ID in the device symbol file (.SYM). This file is located in the ... \DEVICES\HART\[MANUFACTURERID]\[DEVICETYPEID] folder. For instance, our device symbol file would be located here: ...\DEVICES\HART\000026\0006\0602.SYM. This symbol file lists all the parameters known to the device, their names, symbol IDs, and variable types. By cross-referencing the symbol ID in this file with its counterpart in the simulation file, we can determine which parameter we are looking for. Here is a partial listing of the symbol file for our device:

```
variable response_code enumerated 150 (std,mand,imp)
variable hardware_revision unsigned 159 (std,mand,imp)
variable sensor_a_lower_sensor_trim_point float 6004 (std,imp)
```

The drawback to this method is, of course, that the AMS Device Manager-displayed parameter label is nowhere to be found.

B.4 Limitations of a simulated device

While simulation of a device can be a useful technique for device integration, there are some drawbacks to the simulated device.

Non-dynamic reads and writes

AMS Device Manager reads the simulation file once when the simulated device is identified, generally when the Device Connection View is initially displayed. Unfortunately, subsequent reads of the device are not done, so you cannot expect to change a value in the device simulation file and expect it to be updated in AMS Device Manager in real-time, even for dynamic parameters. The same holds true for writing values to the simulated device. AMS Device Manager does not write the value to the simulation file until AMS Device Manager exits.

Methods do not run reliably in simulation

Although you can run methods in simulation, it is not a good idea to use simulation to debug and conclusively test them. Without a live device to produce the correct responses, methods will more than likely not run correctly under simulation.

Local variables not read or written

Some devices use variables of class LOCAL and have actual read and write commands for them. Although this is bad practice, it does work with live devices through AMS Device Manager and the 375 Field Communicator or 275 HART Communicator. It does not, however, work in simulation. These parameters will not be read from or written to the simulation file.

Emerson
12001 Technology Drive
Eden Prairie, MN 55344 USA
T 1(952)828-3032
F 1(952)828-3033
www.Emerson.com