Device Revision and Lifecycle Management Guide

How to keep systems up to date and compatible with new devices using EDDL

1 Device Integration and Compatibility

The challenge of integrating devices and to stay compatible with new types and versions of devices coming into the plant has existed since the very first smart transmitters and other intelligent devices. First DD, and then EDDL, were created specifically to enable interoperability between systems and devices and easily keep the systems up to date with new devices: simple device integration and device revision management. EDDL is the key to interoperability.

1.1 Plant Challenge: Device Revisions

Devices get better and better. The parade of new versions for intelligent devices is relentless and unfolding for all types, manufacturers, and protocols as devices constantly improve. The challenge for a new project and over the lifecycle of an operational plant is managing different versions[1] for devices using digital communication. More than one version of the device may arrive on site, from the manufacturer's different locations, reseller's stock, as part of package units, or as ongoing replacements. Managing all these device types and versions is a challenge. New types of 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART devices and new versions of devices are made available from hundreds of manufacturers every week.

Pure analog 4-20 mA devices are virtually extinct. Systems that use hardwired on-off and 4-20 mA signals for control now use 'smart' transmitters with digital communication like HART for calibration and configuration/setup. That is, all plants need field communicators or laptops, and these too need to be kept up to date with the latest types and versions of devices. Therefore, device revision is a challenge also for 4-20 mA devices.

In past, every time a new device type or version came to the plant the 'smart' handheld had to be sent to the manufacturer for upgrade, and a new version of the configuration software had to be obtained, in order to stay compatible with new devices types and versions.

1.2 Solution: EDDL

EDDL (IEC 61804-3) is a device integration technology created as a solution to the device revision problem. EDDL makes integrating and managing devices of different types and versions easier. EDDL[2] is a file that is loaded onto the workstation or handheld field communicator. There is no EDDL inside the device[3] itself. Thus there is no such thing as an “EDDL device” or a “non-EDDL device”. Devices are 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART devices, and these protocols support EDDL. The same EDDL file is used for all systems, and shows the same device information on all systems (see appendix 3).

If the system is not updated in lock-step with new devices and revisions, it will not be able to fully communicate with these new devices. Even 4-20 mA devices rely on digital communication for configuration and calibration before they can be commissioned. However, the system manufacturer cannot possibly update and test system software as fast as device manufacturers release new devices. The solution provided through EDDL is that the device manufacturer suppliers the EDDL file for their device. Every version of every device type from every manufacturer has its own file. In the plant, the technician then loads the EDDL file onto the system or handheld, the system and handheld are now able to communicate with the new device without making changes to the software.

---

[1] HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART literature use the term “revision” when speaking of different versions. In this guideline the terms “revision” and “version” are used interchangeably.

[2] Technically EDDL is a language used to write an EDD (file) describing a device just like HTML is a language used to write an HTM file describing a web page but common industry practice has become to say EDDL-file just like we say HTML-file

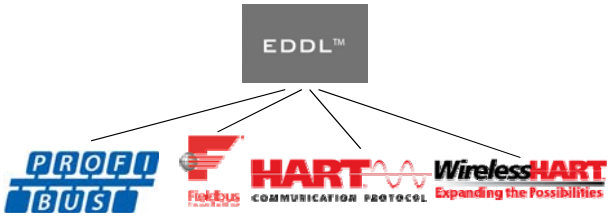
[3] FOUNDATION fieldbus does provide the ability to store the EDDL file in the device so that it can be uploaded in the system, but this feature is currently not used

The purpose of EDDL is to ensure there is no need upgrade software version to match device version

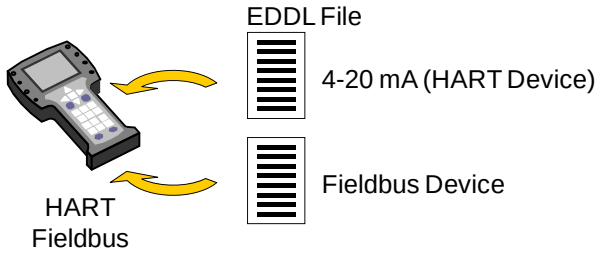
That is, thanks to EDDL the system is kept evergreen and up to date with new devices. The problem of managing device revisions is solved.

1.2.1 Same Technology, Same Procedures

The Electronic[4] Device Description Language (EDDL), formerly known as just Device Description (DD[5]), has been supporting device integration and device revision management since circa 1992. EDDL is a *device integration technology*. EDDL is an integral part of 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART protocols. EDDL is mainly associated with FOUNDATION fieldbus because Fieldbus systems always use Intelligent Device Management (IDM) software while 4-20 mA systems with 4-20 mA devices (configured through HART protocol) rarely had intelligent device management software in the past, most of the time a handheld field communicator was used, often upgraded in factory. However, EDDL is also used for device integration with the other bus technologies whenever configuration/setup, calibration, or diagnostics is done.



The 4-20 mA signal itself works without EDDL, the system gets the process variable, but 4-20 mA devices with HART requires EDDL to be commissioned, for sensor calibration trim, configuration/setup (range), and diagnostics - for instance with a field communicator or laptop which must have an EDDL file loaded.



Once a plant deploys an Intelligent Device Management (IDM) software part of the asset management system, EDDL files will have to be loaded on the system too, not just the handheld field communicator. This holds true for 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART protocols.

1.3 System-Device Backwards and Forwards Compatibility

EDDL ensures compatibility between the system and devices of older and newer versions. EDDL ensures *backwards compatibility*; by loading EDDL files for old devices, new software and handhelds work with old devices. That is, old devices do not go obsolete with new systems. EDDL ensures *forward compatibility*; by loading EDDL files for new devices software and handhelds will

[4] Some literature erroneously state ‘E’ in EDDL means "enhanced", but this is incorrect. Similarly, the abbreviation “eEDDL” for enhanced EDDL should also not be used.

[5] DD is sometimes called "Device Driver" but this is fundamentally wrong because a DD is not driver software. Other times DD is called “Device Display” which is also not correct

work with future devices as they become available without having to upgrade the software. That is, old systems do not go obsolete with new devices.

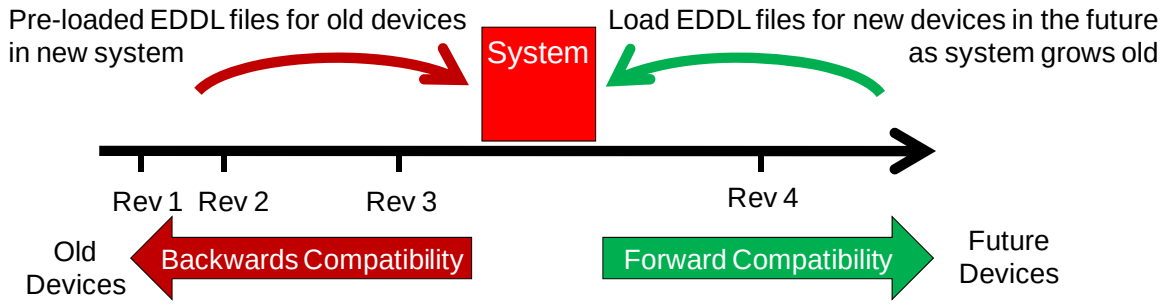


Figure 1 EDDL makes system forward compatible and backwards compatible with devices

Example: The plant purchases the system when a particular device is at revision 3. This system also works with revision 1 and 2 of that device (backwards compatibility) because the EDDL file for all three revision is pre-loaded on the new system. In the future, when the plant purchases that same device type again as a replacement, it may be at revision 4. By loading the EDDL-file for revision 4, the system will work for revision 4 too, even though revision 4 did not yet exist when the system was purchased (forwards compatibility).

1.4 How EDDL Works

When purchasing a system, software, or handheld field communicator it comes with most EDDL files pre-loaded. However, new device revisions become available every month. If a new device is purchased months later, the EDDL file for it must be loaded by the system administrator.

In order for a DCS, intelligent device management software as part of an asset management system, or a handheld field communicator to communicate with a device it needs to know what to transmit, what it receives, and what to display to the technician. The EDDL file for the device provides this function. The EDDL file tells the system which commands to send to the device and how to interpret the response from the device. Thanks to the EDDL file, any system or handheld can communicate with any type of device from any manufacturer and display the information.

That is, keep systems up to date and compatible with new devices by loading the EDDL file for the device onto system

EDDL is the solution to device integration and device revision management in that all versions, old and new, of all device types from all manufacturers can thanks to EDDL be managed from the same software

Upgrading an old system from traditional DD technology to EDDL technology does not require change of skills. EDDL files are copied and pasted just like HART and Fieldbus traditional DD files are copied today, no software installation has to be learnt.

Most 4-20 mA control systems don’t actually use the HART digital communication. The 4-20 mA/HART device is configured using a handheld field communicator at the time of commissioning. Therefore EDDL and other internal system files are only loaded on the handheld field communicator; there is no need to load EDDL and other internal system files on 4-20 mA control systems. Moreover, the handheld field communicator may come with easy-to-use DD-file update management software utility that makes it as simple as a click of a button to update it with EDDL and internal system files. The utility downloads the device files for devices from different device manufacturers from the field communicator manufacturer’s server and load into the communicator automatically. Systems using FOUNDATION fieldbus devices need to be loaded with their EDDL files. Note that the same applies to systems using 4-20 mA/HART devices with intelligent device

management software; EDDL and other internal system files for the 4-20 mA/HART devices have to be loaded.

There is only one DD per device type, from the device manufacturer (not from system vendor, system integrator, or other third-party), and it is identified by manufacturer code, device type code, and DD file revision so system can automatically pick the DD file for any device connected. There is no need for manual assignment of a driver.

1.4.1 System Integration

Because the EDDL file fully describes each and every parameter in a device, the corresponding parameters can be automatically created in the system database. There is no need to manually create individually system tags or manual data mapping parameter by parameter. With hundreds of parameters in each device, EDDL saves countless hours of system configuration work and troubleshooting. The device information is not only displayed in the intelligent device management (IDM) software, but every parameter can be used in the DCS or by external software applications though an OPC server if available. See separate white paper on system integration.

1.4.2 New Skills Required

EDDL files are downloaded from the Internet and loaded on the system. In the past, people in general were not as used to using software, browsing the Internet, downloading files, and synchronizing with devices. As a result, many plants had difficulties managing device revisions because they had difficulties downloading the EDDL file onto their system. However, now MP3 players, digital cameras, smart phones, and tablets have now changed this. Using software, downloading files, and synchronizing with smart devices is becoming second nature. Managing device revisions and EDDL files is easy.

1.4.3 No Other Means to Achieve A Comparable Result

There are other technologies to display setup and diagnostics, but EDDL is the simplest and most robust solution. Other device integration technologies have drawbacks which EDDL avoids.

2 Basics

Device revision management is all about keeping the system and handheld field communicator up to date with the latest devices.

2.1 System and Handheld Update

The EDDL files are loaded into the system's "Device Library". The EDDL file can be obtained from one of many sources: device manufacturer website, HART Communication Foundation website, Fieldbus Foundation website, PROFIBUS International website, or system manufacturer website. Typically it is best[6] to obtain the EDDL file through the manufacturer of the system used (the EDDL file is created by the devices manufacturer, but system manufacturers typically distribute the unmodified EDDL files to their users together with internal system files - see appendix 3). Normally the EDDL files are downloaded from a website, but they are so small they can also be emailed as attachments. Because they are small they can also be downloaded across a poor Internet connection without suffering time-outs. In some case whole "libraries" of multiple EDDL files are provided on CD/DVD on a quarterly basis or through Internet on-line update. EDDL files can be copied and pasted from one system to another. This makes managing device revisions to keep the system up to date using EDDL easy.

The device revision[7] of the EDDL file to be loaded on the system shall match the device revision of the device. When the system or handheld encounters a new device for which it does not have the corresponding EDDL file with the correct device revision, the system will prompt the technician which EDDL file it needs the technician to obtain and load. This makes getting the right file easy.

In A Nutshell

If the plant receives a new revision of a device, obtain the EDDL file for that device revision and load on the system.

For instance, if the plant receives a revision 7 device, get an EDDL file for device revision 7.

The EDDL files are copied into folders and sub-folders for each manufacturer and device type. Most systems have a wizard that automatically adds new EDDL files into its correct folder of the device library. This eliminates mistakes and makes keeping the system up to date easy. In some systems, the EDDL file need only be loaded once on the engineering station, and the add device wizard will automatically propagate the EDDL files to all stations

The communicators and software are the same as for Fieldbus and 4-20 mA/HART devices and they are kept up to date the same way: by loading EDDL files into the device library. If this is not done, configuration and calibration cannot be done, so even a 4-20 mA device cannot be fully commissioned without loading EDDL files onto the handheld field communicator or other configuration tool.

Figure 2 EDDL works the same way for 4-20 mA/HART, WirelessHART, FOUNDATION fieldbus, and PROFIBUS

[6] Systems typically also use internal system files created specifically for each device type by the systems manufacturer for their system for device diagnostic alarm management etc. which can be obtained together with the standard EDDL files for the device (see appendix 3).

[7] There is also something which is called "DD revision" for the EDDL file, but this is largely uninteresting to the system and therefore not stated. In most instances the term "DD revision" is used erroneously (refer to glossary).

NOTE

When integrating a 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS or WirelessHART device into an online control system, download the complete set/package of standard device files and internal system files from the system manufacturer's website. The standard device files on the device vendor or bus organization website may not sufficient for earlier versions of DCS software. Most modern DCS do not have this problem as they are able to auto-generate the internal system files from the standard device files.

Managing device revisions is very simple, but many get confused by "device revision" and "DD revision". The system administrator is responsible for keeping the device library up to date.

2.1.1 Automatic Download of EDDL files

Some host vendors may provide automatic download of EDDL file as a paid service for their host. That is, the system may have the option to periodically check for and download EDDL files for new types and version of devices from a file server. Using such a service, the host system will be prepared the latest available EDDL files, such that when such devices appear on site, the system is ready to integrate those devices.

For instance, a field communicator may come with utility software which updates communicator with its latest device support files. This saves the technician the trouble of searching for files on multiple websites. The utility first downloads the EDDL files onto the PC, and then synchronizes the EDDL files from the PC to portable. The utility treats EDDL files for 4-20 mA/HART, FOUNDATION fieldbus, and WirelessHART devices the same way. All are equally easy.

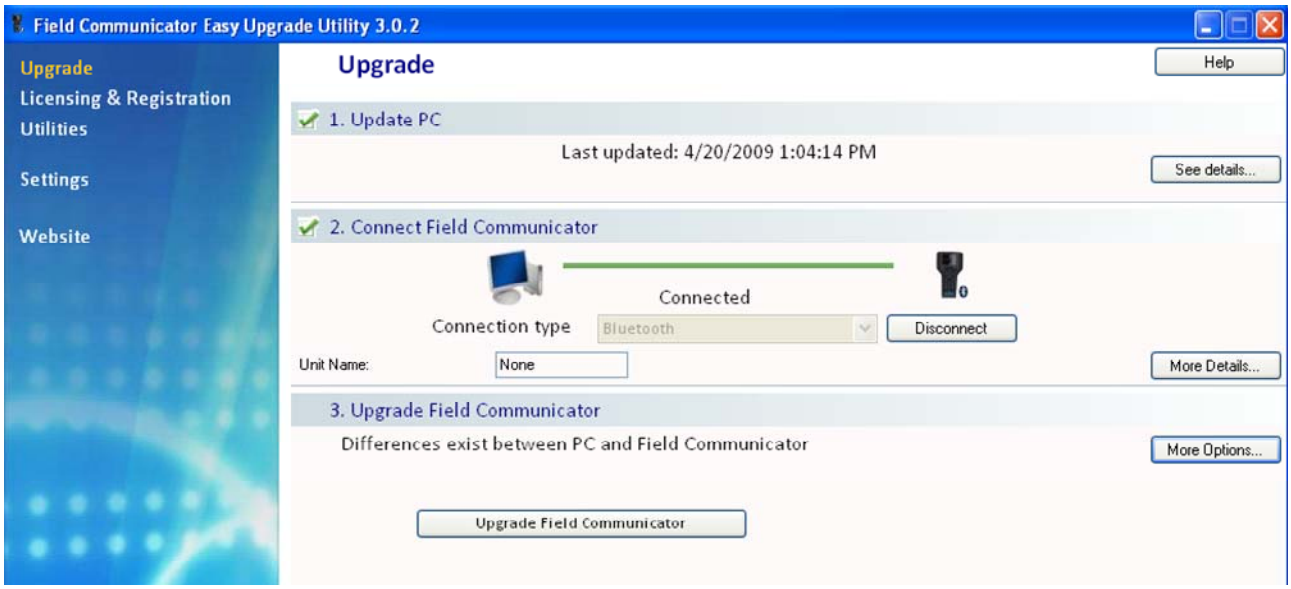


Figure 3 Utility for automatically updating EDDL files on a handheld field communicator

Similar utility software may keep a control and intelligent device management system is up to date automatically, ready for new device types and versions as they come to the plant. This saves the system administrator the trouble of searching for files on multiple websites. The utility first downloads the EDDL files onto a PC, and then synchronizes the EDDL files from the PC to the system either automatically over a network or via manual file transfer. The utility treats EDDL files for 4-20 mA/HART, FOUNDATION fieldbus, and WirelessHART devices the same way. All are equally easy.

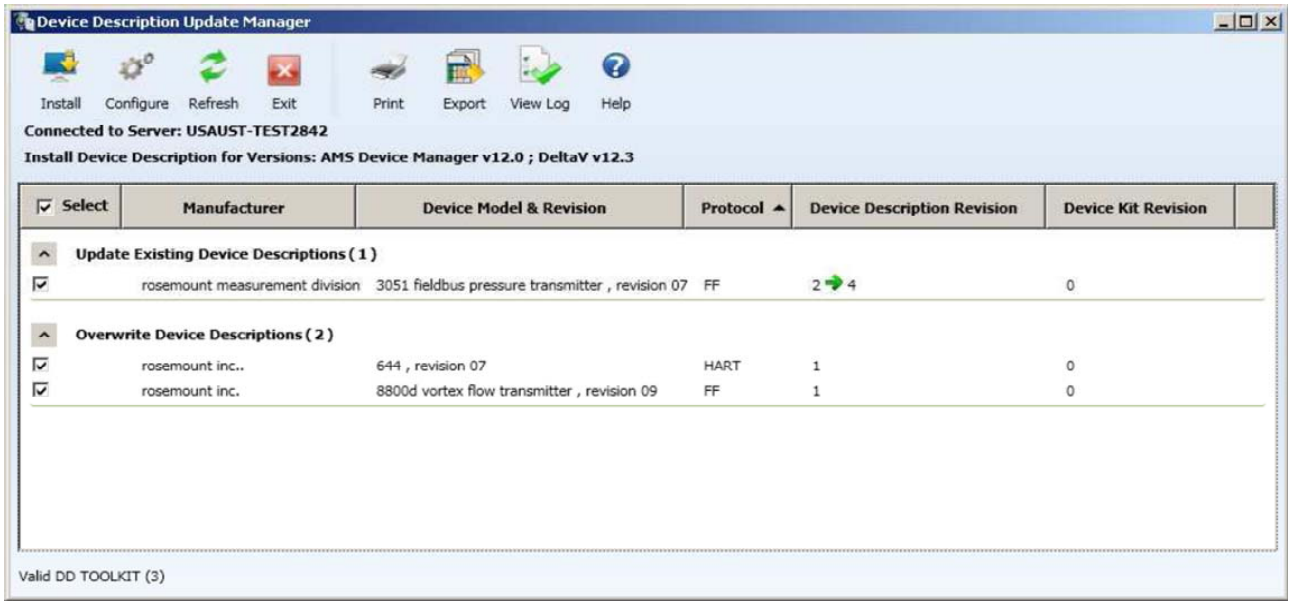


Figure 4 Utility for automatically updating EDDL files on a system

2.2 Device-File Backwards/Forwards Compatibility Access

An EDDL file is created by the device manufacturer for each version of a device. The concept of EDDL is that the EDDL file created specifically for a particular version of a device shall be loaded onto the system. This may not always be possible, for instance when a failed device is replaced, it may unexpectedly be discovered that the replacement device is of a newer version than the failed device and the matching EDDL file may not be on hand or a person authorized to load the file or make necessary changes on the system may not be around (weekend or nightshift). For this reason devices and systems are increasingly supporting backwards/forwards compatibility between device revisions and EDDL files. That is, an old EDDL file is "*forwards compatible*" with the new device (HART terminology), or in other words, a new device is "*backwards compatible*" with the old DD file (FOUNDATION fieldbus terminology).

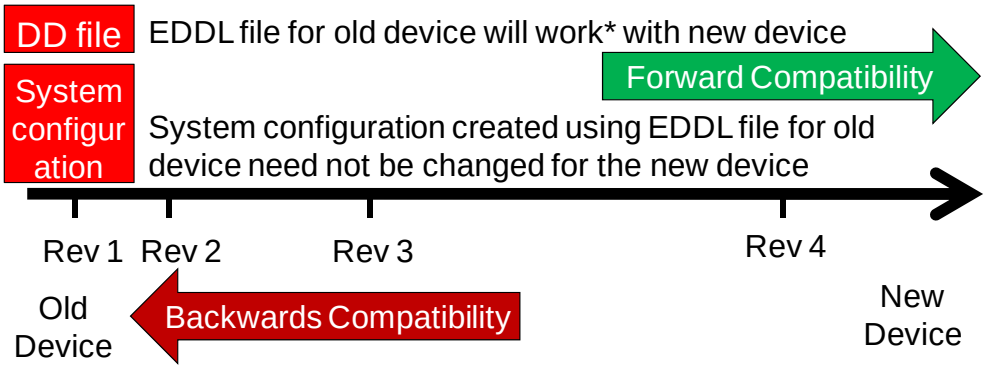


Figure 5 EDDL file for old version device may work with new version device

Note that this backwards compatibility was not possible with FOUNDATION fieldbus devices prior to device profile version (ITK) 6. Therefore device revision management at device replacement was a challenge for FOUNDATION fieldbus devices and systems in the past.

The *compatibility revision* for a device is the oldest version of a device whose EDDL file a newer version device can use. Usually backwards compatibility goes all the way back to the first version so it usually is 1. Should the device manufacturer decide to break backwards compatibility with the earlier versions, the compatibility revision indicate version this was done.

3 Glossary

Just like any technology, EDDL has its own set of terminology and acronyms

3.1 Terminology

Backwards compatibility access	When system does not have the EDDL file matching the device revision of the connected device, it uses an earlier lower version EDDL file in its place. That is, a new device is used “backward” with an EDDL file for an older version device.
Capability File	A file used for FOUNDATION fieldbus devices declaring resources and other characteristics in the device required for off-line configuration.
Capability File Revision	Version of the capability file
Capability Level	A subset of device functionality (such as optional function blocks or other features) that are unlocked based on licensing or hardware configuration etc.
Compatibility Revision	The lowest (oldest) device version for whose EDDL file (and configuration) which a new device can use. An indication of backwards compatibility.
DD-file update management software	Software utility which automatically or semi-automatically downloads DD files from a server onto the system.
DD Revision	Version of the EDDL file itself, the user interface
Device Revision	Version of the device
Firmware/Software Revision	Version of the firmware in the device
“Forward compatibility” access	When system does not have the EDDL file matching the device revision of the connected device, it uses an earlier lower version EDDL file in its place. That is, an old EDDL file is used “forward” with a newer version device.
Generic access	Accessing a subset of the device functionality using universal and some common practice commands (for HART) or the resource block, standard transducer blocks, and standard I/O function blocks (FOUNDATION fieldbus) or using device profile (PROFIBUS)
Hardware Revision	Version of the electronic circuit in the device
Intelligent Device Management	Configuration/setup, sensor calibration trim, and diagnostics etc. of field devices.
Plant Asset Management	Machinery health management, processing equipment management, and intelligent device management

In this guideline the terms “revision” and “version” are used interchangeably.

3.2 Acronyms

CF	Capability File
CFF	Capability File Format
DCS	Distributed Control System
DD	Device Description (old abbreviation and term, now replaced by EDDL-file)
DDL	Device Description Language (old abbreviation and term, now replaced by EDDL)
EDD	Electronic Device Description (same as EDDL-file)
EDDL	Electronic Device Description Language
GSD	Geräte StammDatei ("Device master file")
HART	Highway Addressable Remote Transducer
HIST	Host Interoperability Support Test

HTML	HyperText Markup Language
IDM	Intelligent Device Management
PAM	Plant Asset Management
HPR	Host Profile Registration
LOI	Local Operator Interface

4 Questions

Post your questions on device integration, device revision management, and intelligent device management with the EDDL group on Linkedin:
http://www.linkedin.com/groups?gid=3736433

5 References

EDDL Brochure and Technical Description on www.eddl.org site

Jonas Berge, "Fieldbuses for Process Control: Engineering, Operation, and Maintenance", ISA, 2002, ISBN 1-55617-760-7

NAMUR NE 105, "Specifications for Integrating Fieldbus Devices in Engineering Tools for Field Devices", Version: 24.08.2004

Appendix 1 Versioning and File Names

A new revision device has new features such as new function blocks, new parameters (setup, diagnostics, monitoring), new options (units, sensor types, etc.). The system must know the features of every revision of every device type in order to make all of the features available to the technician.

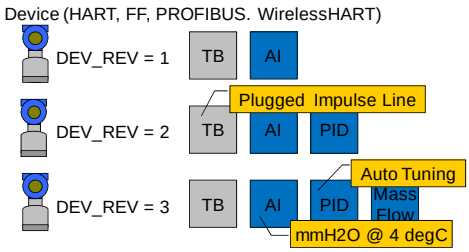


Figure 6 Devices keep getting better and better with new features and options

Every revision of every type of device has its own EDDL file. This is because by definition each revision of a device has some new feature. It must have a new EDDL file so the device manufacturer can describe how this feature is communicated and how it shall be displayed in the system.

1.1 Version Management (Device)

The EDDL file lets the system know the features of each device. Therefore the system requires an EDDL file to be loaded for each device to describe its features.

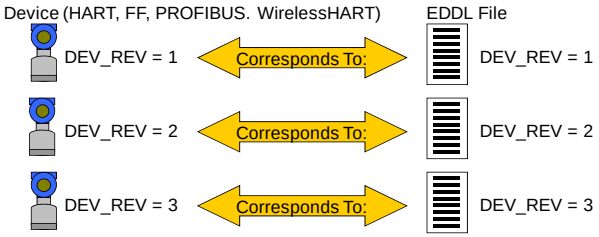


Figure 7 Each device version has a corresponding EDDL file

An important concept in device integration and device revision management using EDDL files is that there is no tampering with what is already working. Existing EDDL files are not removed or overwritten, just add new files. For instance, when a new revision 2 device comes to the plant, the existing EDDL file for device revision 1 is still left on the system. A new EDDL file for device revision 2 is simply added. That is, EDDL files for all device revisions remain on the system. There is no risk of an existing working file being overwritten by a new file which is not working. This ensures a new device does not break the operation of an old device. Old and new devices can operate together in the same system. There is no risk of breaking what is working. EDDL makes device integration and device revision management easy and maintains system robustness.

When EDDL file for a new revision device is added, the existing files are not touched, they continue to work as is. What is running is not upset. Old and new versions can coexist.

As far as device revision management and EDDL files is concerned, device "hardware version", "firmware version", and "software version" are not relevant to the system, the system cares about device revision. For simplicity, device manufacturers should not refer to hardware, software, and firmware version, but only device revision. Usually when there is a new hardware or firmware/software version for the device there is also a new device revision, but it may not always be the case since the hardware or firmware/software change may not be related to communication. That is, do not confuse project documentation by including hardware, firmware, or software version, specify only device revision. Moreover, some devices have a separate communication

board sandwiched together with the main processor board in the device. Device revision reflects the firmware version of communication board, not the main board. Again, specify only device revision in project documentation. Lastly, do not concatenate Device Revision and DD Revision as one number as this will cause confusion with firmware version etc. For instance, if Device Revision is 7 and DD Revision 2, do not call this "7.2" because it will cause some to believe this is a firmware version, software version, or hardware version.

Hardware, software/firmware version information may be useful to a device manufacturer for troubleshooting and for reference to knowledge base articles etc. Hardware, software/firmware version is of interest when the factory is verifying internal compatibility between different circuit boards or compatibility with proprietary software.

1.2 Version Management (EDDL file itself)

The device manufacturer may also choose to create a new EDDL file even though there is no change in the device, for instance to improve the device user interface and guidance with easier to use wizards, more intuitive menus, better graphics, conditionals, help text etc. When there is a new EDDL file for the same device revision, this is called a new "DD revision"

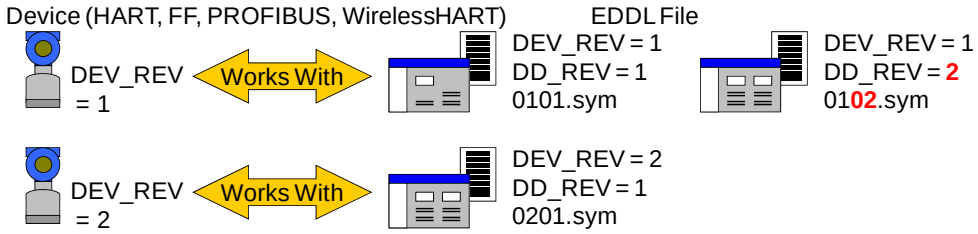


Figure 8 The device manufacturer may improve the user interface without changing the device

Common Misunderstanding

Many assume that DD revision of the file must equal device revision of the device, but that's NOT how it works. "What is the DD revision of the device?" is generally the wrong question.

Note that when device revision in incremented (new version of the device), the DD (file) revision starts over from 1.

Table 1 DD revision restarts from 1 for new device revisions

Device Revision	DD Revision	Remark
01	01	Original
02	01	New device revision
02	02	New EDDL file revision
03	01	New device revision

System software is not concerned[8] with DD revision (it's the device revision which really matters). The system will simply use the highest DD revision[9] loaded on the system.

A higher DD revision only means that the EDDL file has changed. The device itself is not changed, it is exactly the same: same firmware version etc. The device have exactly the same

[8] An exception is Honeywell and ABB systems which, at the time of writing, are sensitive to DD revision which must match

[9] FOUNDATION fieldbus devices have a DD_REV parameter in the resource block, originally intended to state the DD file version desired for the device. Ignore this parameter. It does not make sense, because the DD file is revised independent of the device firmware, so this parameter will not be updated, and therefore is not useful.

blocks/commands, parameters/settings, diagnostics, and options etc. It is only the EDDL user interface such as menu structure, graphical representation, and wizards that has changed.

In summary, a device does not have a DD revision; the device has a device revision. DD revision is the version of the file. Therefore DD revision should not be stated in instrument specification sheets when purchasing.

There has been some confusion on DD revision and device revision in the past. Therefore some documents erroneously state "DD revision" although it is evident from the context they really mean device revision.

Table 2 User interface version is independent of device version

Device Revision	DD (file) revision
Version of the device	Version of the user interface
Incremented when device firmware is improved with new features: • New blocks • New parameters • New options • New diagnostics	Incremented when the user interface is improved with new features: • Better graphics • Smarter wizards • More helpful conditionals • More intuitive menus

If device firmware will be upgraded, make sure to have the new EDDL file and any internal system files (see appendix 3) on hand first.

The Device Revision for the EDDL file must match the device revision of the actual device, otherwise the system cannot work with the new device. This applies to 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART alike. The system does not care much about the DD (file) revision; it picks the latest available EDDL file for the device.

Device revision is associated with new firmware versions, DD revision is not.

1.3 Understanding EDDL File Names

The EDDL file names consist of two numbers:

- Device revision [2 digits]
- DD (file) revision [2 digits]

Note that these numbers in the file names are in hexadecimal[10]

The advantage of this naming scheme is that each version of the device (and the user interface) has an individual file with a different name. When the file for a new version device is added, the files for older versions remain untouched. Files are not replaced or removed. Old and new devices can coexist in the system. Working files are not removed, established relationships are not broken. The file naming convention (and folder hierarchy) enables the system to automatically pick the correct EDDL file for a device, without having to manually associate a new device with its file.

Table 3 DD file nomenclature

	Example 1	Example 2
File Name	0701.sym	1703.sym
Interpretation	07 = Device Revision 7	17 = Device Revision 23 (17 in

[10] Note that it is not correct to say that 0203 is rev 2.3 or that 0301 is rev 3.1

	01 = DD File Revision 1	hexadecimal is 23 in decimal) 03 = DD File Revision 3

Each protocol use a different file extension for the system software to identify them but as far as humans are concerned the files are all managed the same way. This makes managing EDDL files consistent and easy. In fact, the file extension is a good way to tell if the file is for the 4-20 mA/HART protocol or FF protocol version of the same device.

The DD file format has evolved over the years. However, DD files are distributed as a ZIP file that contains multiple DD files including the latest (enhanced) EDDL format as well as old (traditional) DD format so it can be used on both older and newer systems.

It is interesting to note that internal system files for devices also use the same file naming convention, simply identifying the file type by a different extension.

1.3.1 4-20 mA/HART

HART protocol files are identified with the following extensions:
*.fms (Traditional)
*.sym (Traditional)
*.fm6 (Enhanced)
*.fm8 (Enhanced)
*.fma (FDI)

With *.fm6 and *.fm8, a separate *.sym file is no longer required

1.3.2 FOUNDATION fieldbus (FF)

FF device files are identified with the following extensions:
*.ffo (Traditional)
*.sym (Traditional)
*.cff
*.ff5 (Enhanced)
*.sy5 (Enhanced)
*.fm6 (FDI)

'Incremental DD' is a scheme of using multiple partial DD files. However, multiple DD files are more difficult to manage for both device manufacturers and system administrators so this scheme is not recommended.

1.4 Understanding Capability File Names

The FF device capability file names consist of:

- Device revision [2 digits]
- DD file revision [2 digits] – not relevant
- CF revision [2 digits]

Note that these numbers are in hexadecimal

CFF revisions are not coupled to DD file revisions. Therefore the middle DD file revision number is not relevant.

Table 4 Capability file nomenclature

	Example
File Name	070101.cff
Interpretation	07 = Device Revision 7

	01 = (DD File Revision 1) 01 = Capability File Revision 1

Capability file revisions and DD file revisions are updated independently. There need not be a new capability file revision with every new DD file revision (there usually isn't). A new revision device may be supplied with the same capability file as before. Thanks to the naming scheme the system knows which file to use. Note that when device revision is incremented (new version of the device), the capability file revision starts over from 1 only if a new capability file is required/supplied for the new device. However, depending on the changes, the new device version may not need a new capability file and may therefore be supplied with the same earlier capability file. However, when DD (file) revision is incremented (new version of the EDDL file), the capability file revision does not start over. That is, the earlier capability file continues to be used with the new device's new EDDL file.

Table 5 Previous capabilities files can be used newer DD files

Device Revision	EDDL File name	CF File name	Remark
01	0101	010101	Original
01	0101	010102	Improved capability file (earlier EDDL file still used)
01	0102	010102	Improved EDDL file (earlier CF file still used)
02	0201	010102	New device version (new EDDL file, earlier CF file still used)
02	0201	020102	New capability file (earlier EDDL file still used)

For instance, version 7 of a device may be supplied with a DD with files named 0703.sym and 0703.ffo, and CF file named 070206.cff.

If a device manufacturer releases a new capability file, the file name will reflect the device revision for the device and the DD revision for the DD file. For a new device revision, the capability file revision will always be 01. Some device manufacturers release new DD file (higher DD revision) without creating a new capability file in case there are no changes required to the capability file. A device "file package" may therefore include a DD file named 0203.ff5 and a capability file named 020102.cff. The DD revision between the two files may not be matching, but most systems understand and accept this. If not, consider copying the old capability file and give the copied file a name with matching DD revision.

1.5 Understanding EDDL Folder Names

Software and handheld field communicators store EDDL files in folders and sub-folders for each protocol, each manufacturer, and each device type. Systems have wizards that automatically copy the EDDL files into the correct folder.

If need be, it is relatively easy for a human to find the location of an EDDL file associated with a particular device on the workstation. In the workstation file system, each manufacturer folder is identified by a hexadecimal number (6 digit manufacturer ID, see appendix 6). The manufacturer folder in turn has sub-folders for each device also identified by hexadecimal number (4 digit device type), inside which the EDDL files are stored. The system software has a device library that displays all the available manufacturers and device types as plain text, i.e. names and model numbers recognizable by humans. The manufacturer IDs are not the same for the different protocols (see appendix 6), but for the system administration purposes this does not matter since system software's device library utility shows manufacturer names in plain text, not as ID.

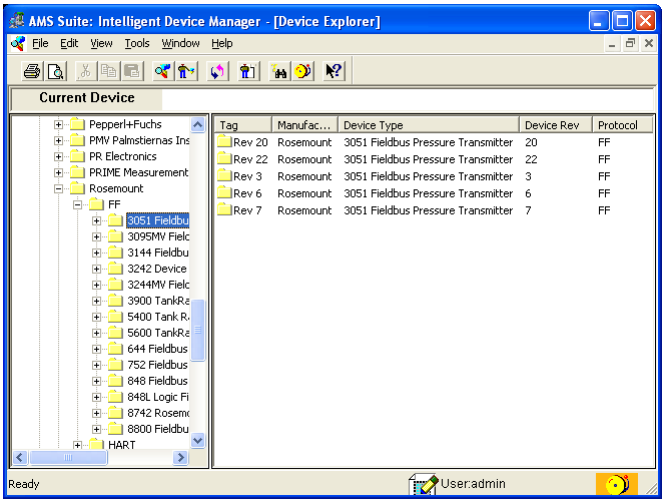


Figure 9 In intelligent device management software devices appear in plain text rather than codes

The folder hierarchy (and file naming convention) enables the system to automatically pick the correct EDDL file for a device when the new device is connected, without having to manually associate a new device with its file.

It is interesting to note that internal system files for devices are stored in the same folders and also use the same file naming convention, simply identifying the file type by a different extension.

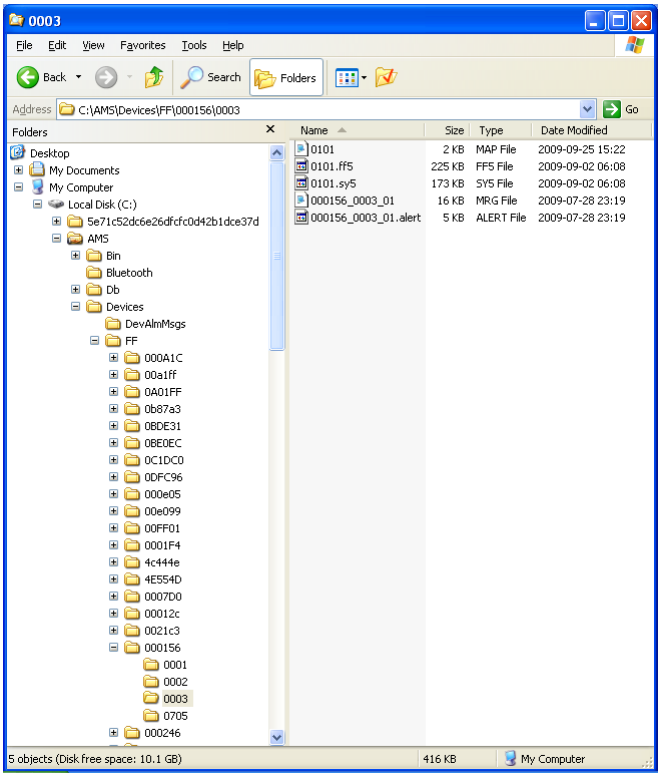


Figure 10 In Windows Explorer files and folders appear with thier coded names

Note that since EDDL files are compressed text, not software, they are copied like documents, not installed like software programs. This has many benefits. First, 'administrator' level password need not be given to every technician that need to be able to commission a new type or version device. This has a cyber security advantage since lower level passwords can instead be given, for which software cannot be installed on the system. The second benefit is that conflicts with drivers and programs are avoided as DLL components and other executables are not installed and no Windows registry changes are made. EDDL files can easily be removed completely if need be. That is, since

an EDDL file is text, not software, this makes it very much easier for the system administrator to support a system based on EDDL.

1.6 Plug-'n'-Play

EDDL has auto-discovery to automatically associate all the hundreds or thousands of devices in the plant with their respective EDDL file. The system hierarchy is automatically populated with the associated devices with the right file from the right folder. This eliminates a lot of configuration work to associate a new device with its EDDL file. The system auto-detect devices and read manufacturer ID, device type, and revision, then automatically associate the device with a particular EDDL file, which is automatic since the EDDL file is named according to the version and is located in a folder and sub-folder named corresponding to manufacturer ID and device type. That is, not only automatically detect a device but to also automatically associate the device to its EDDL file; plug & play.

Grappling with a second device integration technology in addition to EDDL comes at a high price of both time and money. Using only EDDL avoids these issues reducing cost and saving time.

Appendix 2 Device Revision Incompatibility Troubleshooting

EDDL files themselves do not fail, they are compressed text files, not executed software programs. Also, since they are copied onto the system, not installed like driver software, they do not cause DLL and other executable software conflicts or Windows registry issues. However, the software that use EDDL files may show prompts related to device integration and device revision management (EDDL files).

2.1 DD File Missing

If a device is connected to the system or handheld field communicator for which the EDDL file has not been loaded, the system will tell the technician the device revision so the technician can go get the right EDDL file to load. That is, if a device with device revision 02 is connected and the EDDL file for device revision 02 has not been loaded, an error will be displayed.

A FOUNDATION fieldbus device can be detected but cannot be commissioned without its matching[11] EDDL file. Make sure to have the EDDL file and internal system files available before upgrading or replacing a device.

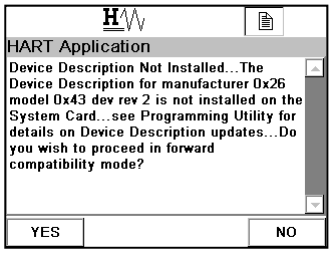


Figure 11 When EDDL file is missing system notifies which file is required

Similarly, the analog signal of a 4-20 mA/HART is available even if there is no EDDL file, and it may be possible to partially configure the device through generic access or "forward compatibility mode". However, complete setup, sensor calibration trim, and detail diagnostics is not possible. Therefore, make sure to also have the EDDL file and internal system files available for 4-20 mA devices before upgrading or replacing the device.

Use automated or semi-automated DD-file update management software to keep the system and handheld communicator up to date with files for the latest device types and revisions, avoiding this problem.

2.2 Internal System File Missing

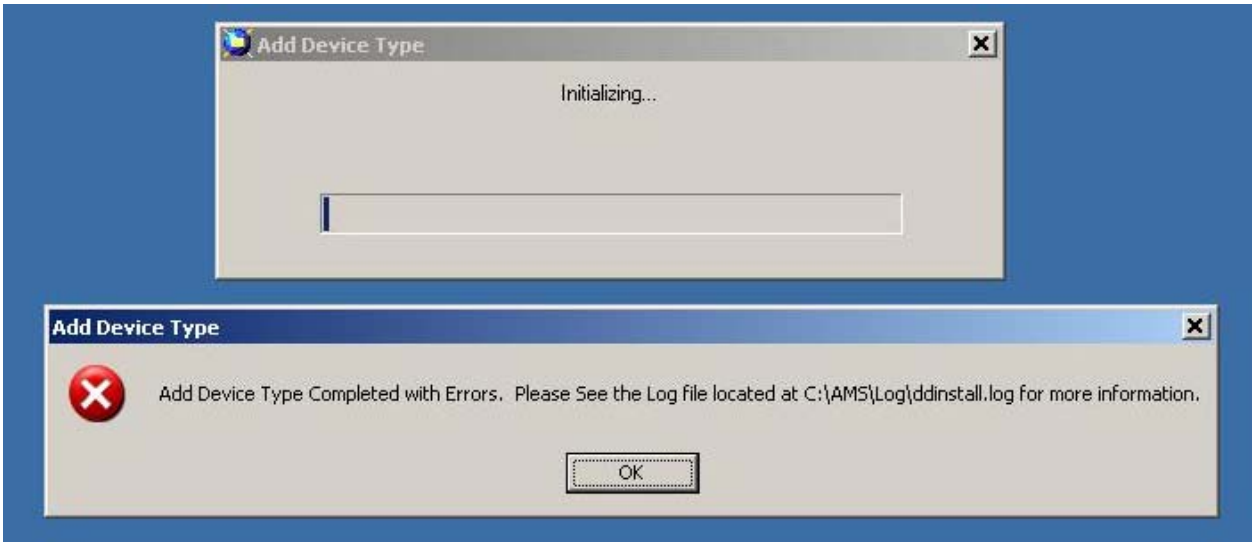
The system will prompt if some internal system file is missing. Modern systems are usually able to auto-generate their internal system files from the standard device files from the device manufacturer or bus organization website. Similarly, a handheld communicator may come with a software utility which generates the internal system files from the standard device files.

Newer systems don't have this problem. Upgrade to a version which automatically generate templates and other internal system files

2.2.1 Internal System File Generation Error

The internal system file generation mail fail, generating an error.

[11] Some systems and devices support 'backward compatibility' / 'forward compatibility' in which case Device Revision need not match; a new device can be commissioned with and old DD file



2.3 Unable to Interpret CFF

The system will prompt if it is unable to interpret the CFF file. Obtain modified CFF from system supplier, or edit the CFF file.

Newer systems don't have this problem as CFF handling is tested as part of the Host Profile Registration (HPR). Upgrade to a system version which has passed HPR.

2.4 Device Revision Mismatch Error

If the system database tag (place holder) is created for a device revision different from the device revision of the instrument connected, the system will not permit the device to be commissioned to that tag. The database configuration has to be modified. For instance, if the system database tag was created for device revision 02 but a device with device revision 03 is connected, it cannot be commissioned.

For a FOUNDATION fieldbus device upgrade, the DCS configuration has to be modified for the new device revision. Depending on the system this can be a simple operation. The intelligent device management software in the system will generally not require a configuration change since it just reads data online.

For 4-20 mA device upgrade or replacement the DCS configuration generally need not be changed as the DCS does not use device-specific HART features. The exception may be smart logic solvers which monitor device type changes at replacement to maintain the integrity of the safety instrumented function. The intelligent device management software in the system will generally not require a configuration change since it just reads data online. However, the EDDL and internal system files for the 4-20 mA/HART is required in the intelligent device management software or handheld field communicator to enable it perform configuration and calibration before the 4-20 mA device can be commissioned.

2.5 DD Revision Mismatch

This error should not be happening. The system should accept[12] a DD file as long as the device revision is matching, regardless of DD revision.

Newer systems don't have this problem as DD revision handling is tested as part of the Host Profile Registration (HPR). Upgrade to a system version which has passed HPR.

[12] An exception is Honeywell and ABB systems which, at the time of writing, are sensitive to DD revision which must match

2.6 Device is not detected

EDDL files are not required to detect the presence of a device. If a device is not detected (2nd level communication problem), it is not a problem with EDDL. If the EDDL file is missing, the device will be detected, but the parameters will not be displayed, or it will be displayed with a limited generic access (3rd level communication problem).

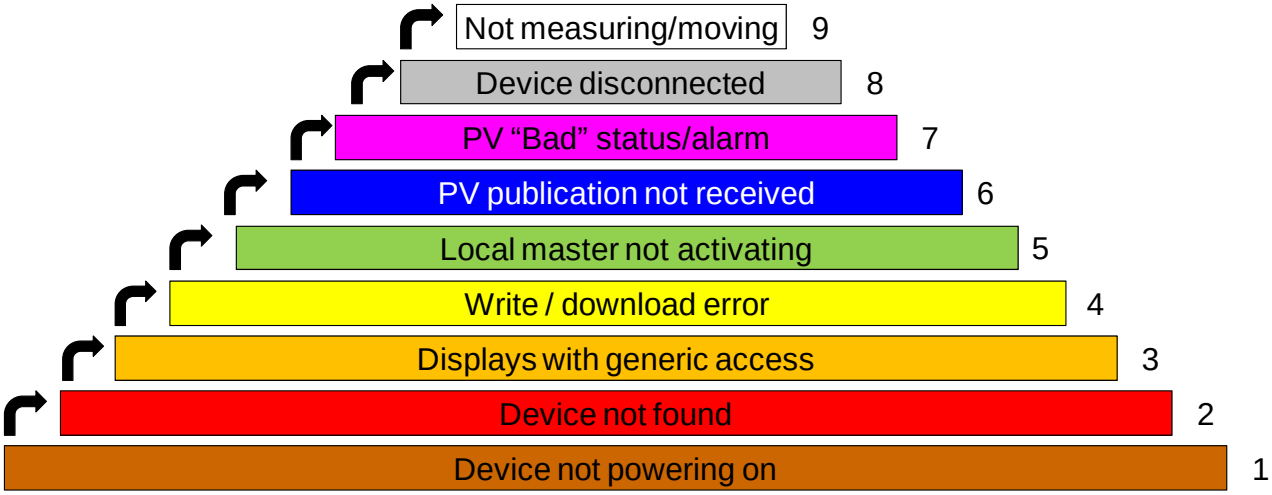


Figure 12 Hierarchical ladder of communication problems

EDDL is only required to access the configuration of the blocks and parameters in the device like diagnostics and setup. The system is able to detect the presence of a device without requiring EDDL.

A gateway can join a WirelessHART device to the network without using EDDL provided the network ID and join key have been set. If device refuse to join, it is not because of EDDL file. However, software or a handheld field communicator requires DD files in order to configure the network ID, join key, and update period in the device before it can be detected by the gateway,

2.7 Communication failure

If communication was working before, but stops working, it is not DD problem because DD does not stop working.

2.8 Generic Access

If no EDDL file for the 4-20 mA/HART device is available, the software may provide generic access to the device. This means that some basic functions of the device can be accessed. These are functions associated with universal commands and some common practice commands. The information is presented with generic graphics. This may include change range, unit, and damping, as well as check on transfer function, write protection status, and sensor limits plus running self test and loop test (fixed current forced output simulation) wizards. Basic process variable monitoring is also possible. Generic access might be particularly challenging for non-transmitter type devices such as valve positioners since the generic access interface tend to be skewed towards transmitters.

A HART version 5 or 6 system is unable to identify a HART version 7 device and therefore will display it with generic access. See separate section on HART version 7.

Using generic access it is possible to monitor the process variable in engineering unit and percentage of range, as well as the output current.

The generic access provides identification information including manufacturer name, hardware and software revision, descriptor, message, date, final assembly and sensor serial numbers, tag, Device ID, universal and device specific command revision.

Polling address and burst mode can be configured using generic access. Write protection status, number of requested preambles, and physical signal type can be checked.

It is possible to configure range, unit, and damping. Transfer function and analog output device alarm (high or low burnout) can be checked along with sensor limits including minimum span.

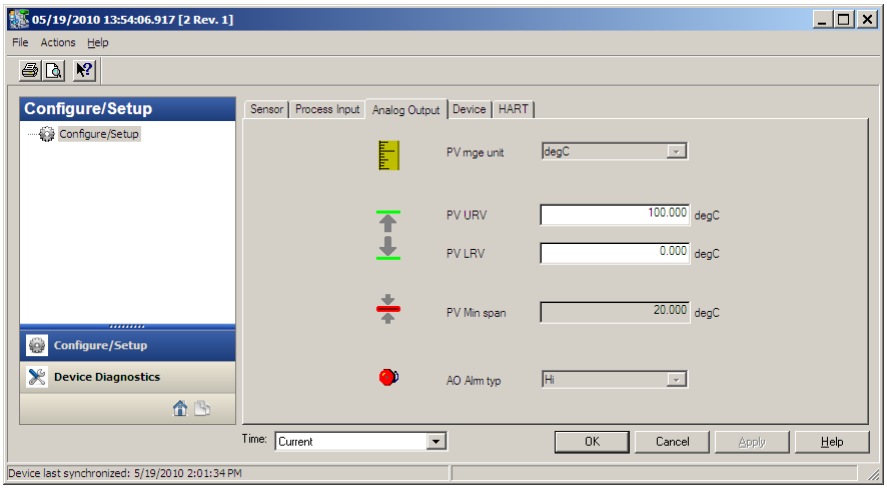


Figure 13 Very basic setup, diagnostics, and monitoring is possible without EDDL file

Wizards available for generic access includes output current trim, self diagnostics test, and loop test (fixed current forced output simulation).

Generic access diagnostics is limited to PV out of limits, non-PV out of limits (for instance senor temperature), analog output saturated (below 4 mA or above 20 mA), analog output fixed (simulation loop test), cold start (power lost and reapplied), configuration changed, and general device malfunction.

Sensor calibration trim, setup/configuration (temperature sensor type, radar level echo curve, flowmeter K-factor, valve type, and valve stroking etc.), and diagnostics requires EDDL file to be loaded to commission the device. Therefore make sure to always obtain the EDDL file for new 4-20 mA device types and revisions.

2.9 "Forward Compatibility Mode"

If the correct device revision of the EDDL file for a device is not available, the technician may be given the option to operate the device in "forward compatibility mode". This means the system will use an EDDL file for an earlier device revision to communicate with the device. Not all features of the new device will be accessible.

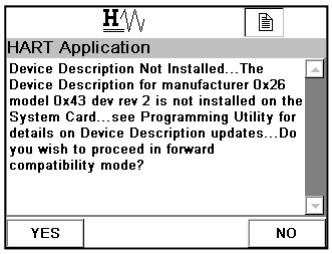


Figure 14 When EDDL file is missing system may provide an option to display in "forward compatibility mode"

2.10 HART version 7

HART version 5 (HART5) and HART version 6 (HART6) compatible systems are not compatible with HART version 7 (HART7) devices. HART5 and HART6 systems will report HART7 as

“unknown device type” and work with “generic” access to a limited set of functions. Some HART7 devices can be configured to operate as a HART5 device. That is, the same device can communicate either HART5 or HART7; and they are technically two different device revisions since the HART5 and HART7 have different functionality, For example, when configured for HART5 the device may be identified as device revision 1, and when configured for HART7 the device may be identified as device revision 2. The device therefore has two corresponding EDDL files. Therefore make sure to load the EDDL file for the device revision corresponding to the HART version of the device which will be used.

Device manufacturers may supply two DD files for their 4-20 mA/HART device. One DD file for the HART5 or HART6 version, and another DD file for the HART7 version. The reason is that HART5 and HART6 systems are not compatible with HART7 devices, so device manufacturers make two versions of the devices available; one HART5 or HART6 version, and another HART7 version, or device can be configured to operate as HART5 or HART7. Depending on the protocol version, the device will have a different device revision. For instance, the HART5 version of the device may be device revision 5, while the HART7 version of the device may be device revision 6. Therefore, the system in this example has to be loaded with the DD file corresponding to device revision 5 or 6 respectively.

Appendix 3 Internal System Files

EDDL files are device specific, written by the device manufacturer according to the IEC 61804-3 standard. Device configuration and diagnostics software (connected ad-hoc as and when desired) does not require any additional files. The device manufacturer EDDL files are sufficient for the software to interact with the device. All the sophisticated configuration information and advanced diagnostics in the device can be accessed using only EDDL

However, a control system or intelligent device management software have functions over and above sophisticated configuration and advanced diagnostics, requiring information that may not be available in the EDDL file (and not any other device integration technologies either). This additional information may include device diagnostic alarm management, operator station device faceplates, or templates with device default configuration values. This information is included in a separate internal system file specific to the system, provided by the system manufacturer (or the device manufacturer in cooperation with the system manufacturer). The information in these files is different for each system manufacturer. The information is also different depending on the device protocol being HART, FOUNDATION fieldbus, PROFIBUS, or WirelessHART, but internal system files are required for all protocols

That is, over and above sophisticated configuration and advanced diagnostics with EDDL (or other device integration technologies), most systems use additional files which are internal and system specific (non-standard). In this case, system suppliers additionally provide the internal system files for each device type. Therefore, for some systems it may be necessary to download these internal system files for a device from the system manufacturer's web site, particularly for older versions of systems. This applies for 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS and WirelessHART devices alike. Note that these additional internal system files are not EDDL files. A "package" of files (zipped) typically also includes the original unmodified EDDL file from the device manufacturer, as well as the capabilities file (CFF) for FOUNDATION fieldbus devices or the device master file (GSD) file for PROFIBUS devices. While the EDDL files for new versions of a device can be downloaded from the bus organization's or device manufacturer's web site, any additional internal system file required come from the system supplier. Most modern systems now have the capability to auto-generate the internal system files they need from the standard device files. This makes it possible to use the standard DD files downloaded from the manufacturer or from the respective bus organization web site.

NOTE

When integrating a 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS or WirelessHART into device an online control system, download the complete set/package of standard device files and internal system files from the system manufacturer's website. The standard device files on the device vendor or bus organization website may not be sufficient for earlier versions of DCS software. Most modern DCS do not have this problem as they are able to auto-generate the internal system files from the standard device files.

As a general observation, system manufacturers can make the internal system files for their own new device types and versions ready quicker "in house" than they can for third-party 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS and WirelessHART devices where obtaining and understanding a test device takes a longer.

One criterion to consider when choosing a control system is to review the system vendor's procedures for development of system support of third-party 4-20 mA/HART, FOUNDATION fieldbus, PROFIBUS, and WirelessHART devices.

Note that there is no difference in the DD/EDDL file for “System-A” or “System-B” because the DD/EDDL files are unmodified. The difference in the file package from one system to the other is the internal system files.

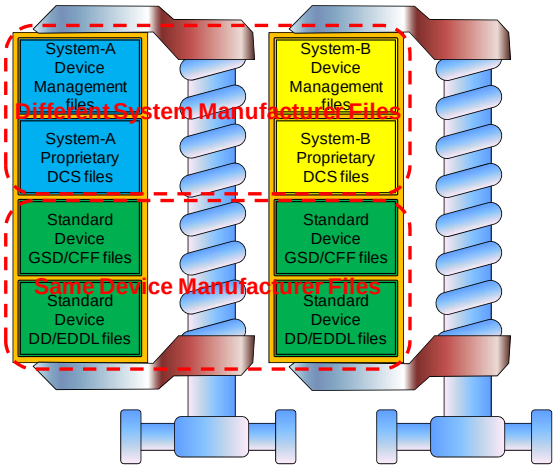


Figure 15 The difference in the file package from one system to the other, is the internal system files

The internal system files should be generated (automatically or manually) and ideally tested against new device types. Therefore "type test" of the system together with these devices and the internal system files may be necessary to check that the system-specific internal system file created by the system manufacturer for the device is matching the device or if it needs fixing (not change the device or the EDDL file, which are already tested by the bus organization). DCS manufacturers may list which devices their system has been tested against - this may sometimes be referred to as HIST[13].

It is interesting to note that internal system files for devices are stored in the same folders and also use the same file naming convention, simply identifying the file type by a different extension.

If the firmware of the device will be upgraded, make sure to have the new internal system files on hand first.

Table 6 Differences between standard device files and internal system files

Device Files	Internal System Files
Standard (IEC 61804-3)	Non-standard
Used by all systems	Specific to each system
Created by device manufacturer	Created by system manufacturer, or auto-generated from standard device files
This is the EDDL files	Other files
Supports device functionality	Supports system functionality • Proprietary system device faceplate (before EDDL) • Device diagnostics alarm management • Default/template settings • etc.
Download from bus organization site or device manufacturer site	Download from system manufacturer site. May also include unmodified EDDL file as a single package

[13] HIST really refers to an early form of control system testing done by the Fieldbus Foundation. Some use the abbreviation to refer to DCS manufacturer testing of field devices. The Fieldbus Foundation HIST has since been replaced by the Host Profile Registration (HPR).

Systems that use FDT/DTM requires the same internal system files for the same reason and additionally also require the device DD files.

3.1.1 Device Graphics and Operator Station Device Faceplates

Traditional DD did not support graphics, this applied to 4-20 mA/HART, FOUNDATION fieldbus, and PROFIBUS devices alike. Therefore some systems added their own device faceplate graphics through proprietary graphics files. Some systems still use such files to display device faceplates integrated in the DCS operator console. These proprietary graphics files must be developed and tested for new device types. Therefore system testing of devices is necessary for such systems. Internal system files for device graphics may be dependent on system versions. EDDL with graphical enhancements eliminates the need for proprietary graphic files thus simplifying device integration and device revision management.

3.1.2 Alarm Monitor Device Diagnostic Alarm Management

An online intelligent device management software part of an asset management system would typically include continuously running “device monitor” software monitoring all devices, which is independent of EDDL (or any other device integration technology). Such device diagnostics alarm monitoring software is a system function that runs in the background on a server at all times even if a technician does not sit at the workstation. It is a mechanism for device diagnostic alarm filtering and routing based on alarm classification and prioritization which is different in every system although the NAMUR NE107 recommendation is increasingly being adopted. The device diagnostic alarm monitor software require information that may not be available in the EDDL file (and not any other device integration technologies either). This additional device diagnostic alarm management information is included in a separate file.

The 4-20 mA/HART protocol does not support device diagnostics alarm prioritization in the field device, it would have to be done in the system device diagnostic alarm monitor. The prioritization/classification of device diagnostics alarms is stored in additional internal system files. Although the FOUNDATION fieldbus protocol does support device diagnostics alarm prioritization in the field device, many system do not support this yet and instead do device diagnostics alarm prioritizations in the system (not in the device).

3.1.3 Device Configuration Template Files

Device configuration template files are used to speed up bulk configuration of large numbers of devices on new projects. The device template can contain the preferred generic configuration settings for a device type, including the NE107 device diagnostic alarm categorization.

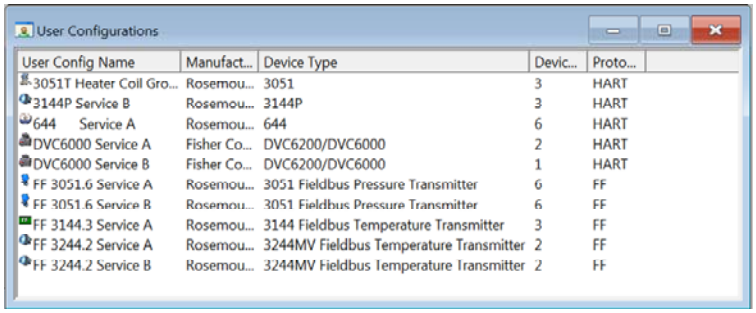


Figure 16 Device configuration templates for each device type and application

Currently there is no standard format for device configuration template files. This means each system uses a different configuration template file format so a system cannot be copied from one system to another. There is some ongoing standardization work in this area. Some systems may not be able to commission a new device without the corresponding template file. Newer system may auto-generate the template files and other internal system files required to commission a device, thus ensuring that delay and downtime is not caused at device replacement.

Appendix 4 Poor Hosts

Original DD technology from 1992 made it possible to integrate and access different types of devices from different manufacturers using the same handheld field communicator or laptop software. Before DD only proprietary device integration solutions existed.

The most basic content of a DD file is the *Device Definition* describing the blocks and the parameters in the device, including limits, options, and help etc. The original DD technology also included *Business Logic* such as "wizards" (aka "methods") which is a kind of script created by the device manufacturer to guide the technician through more sophisticated procedures like setup, sensor calibration trim, or diagnostics (see separate white papers on calibration, device diagnostics, and setup). Wizards and conditionals thus make using devices easy. However, not all intelligent device management software supported wizards. That is, on many systems, user guidance for advanced diagnostics and setup in the past was not so easy, particularly for FOUNDATION fieldbus devices which back then had to set the correct block mode, in the correct block, write value to correct parameter, and remember to return block mode. At the same time the technician also had to remember to inform operations. All of this in addition to the actual task itself such as verifying valve moment or applying an input etc. Lack of wizards was not a problem with the DD technology itself; it was poor implementation in many early products.

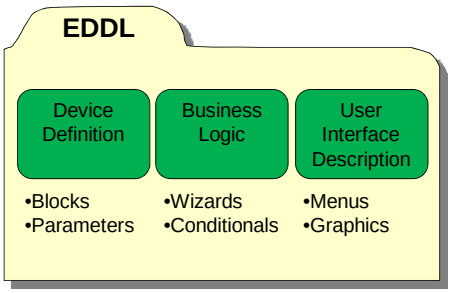


Figure 17 Not all systems supported Business Logic, and the User Interface Description was not introduced until 2006.

Since not all systems supported the full capability of DD, device manufacturers had to provide a simplified DD file with just the *Device Definition* for these simpler systems, without *Business Logic* such as wizards and "conditionals" so that these limited systems could work. That is, the system manufacturer did not modify the DD file. It was the device vendor that provides the simplified DD file.

The EDDL enhancement done in 2006 is a *User Interface Description* which includes graphics such as trend charts, waveform graphs, dial gages, bargraphs, bar charts, and tables etc. Graphics, menu system, wizards, and conditionals are now mandatory for all FOUNDATION fieldbus and 4-20 mA/HART compliant host systems.

In the past, some DCS manufacturers have modified the CFF file removing features in the CFF file not supported in the system. However, most systems simply ignore unsupported features and do not generate errors, thus making work smooth. DD files are unmodified.

Appendix 5 Project Execution Considerations

EDDL simplifies the project execution phase of the system lifecycle including for instance system integration and device commissioning.

5.1 Design: System Specification

In the design phase, make sure the system specification explicitly states that the system software shall render device displays based on Electronic Device Description Language (EDDL) files with enhancements according to IEC 61804-3 for HART, FOUNDATION fieldbus, and PROFIBUS. Shall support DD version 5.1 'device level access' for FOUNDATION fieldbus. Detail device diagnostics display shall open also in operator console when device diagnostic alarm is selected.

Table 7 System EDDL support

Graphical Elements	Menu System
• Image • Multiple pen trend chart • Gauge, Bar graph • Histogram • Bar chart • Multiple waveform graph • Table	• Tabbed card • Hierarchical navigation tree • Pop-up window • Frame

Grappling with a second device integration technology in addition to EDDL comes at a high price of both time to obtain and manage additional files and money for license fees etc. Using only EDDL avoids these issues reducing cost and saving time.

5.2 Design: Instrument Specification

In the design phase, make sure the device specifications for HART, FOUNDATION fieldbus, and PROFIBUS devices explicitly state that device shall be supplied with Electronic Device Description Language (EDDL) file with graphical enhancements according to IEC 61804-3. EDDL files for FOUNDATION fieldbus devices shall utilize device level access (DD v5.1). The EDDL file for the device shall include user guidance in the form of wizards based on EDDL methods for complex procedures, guidance for device diagnostic alarms, plain text labels instead of parameter names, help text, as well as explaining illustrations where applicable.

A project where device variety is managed with MAC/MIV agreements has less than 30 different device types, but if device mixture is not managed the project may end up having many more device types making initial device integration as well as device revision management over the system lifeycle more challenging.

5.3 System Implementation

Before system configuration work starts, the system supplier can request the device manufacturers to confirm which device revision they will be shipping. The system software does not require input of DD (file) revision or CFF revision, thus there is no need to obtain such information. Then build the system database according to these device revisions. It may be a good idea to consolidate all device types in a checklist to easily verify that EDDL files for all HART, FOUNDATION fieldbus, and PROFIBUS devices have been received and loaded on the system. Typically the project documentation already includes a list of different device types, their manufacturers, and protocol. Just add a device revision column to it.

Table 8 Example device revision schedule

Protocol	Manufacturer	Model	Description	Device Revision	Remarks
4-20 mA/HART	Yokogawa	YTA320	Temperature	3	SIS
4-20 mA/HART	Metso	VG800	Valve Guard	6	SIS
FF	CSI	9210	Machinery Health Transmitter	3	
FF	Rotork	FF01 Mk2	Electric Actuator	2	
WirelessHART	Rosemount	3051S	Pressure	1	
WirelessHART	Siemens	TF280	Temperature	1	
PROFIBUS-DP	Siemens	MM440	Drive	2	MCC

Make sure to load the EDDL files for all the device manufacturers the system supports, not just the device types and device revisions believed to be part of the project. The reason being that at the time of plant pre-commissioning it is almost always discovered that there are other undocumented devices at site, for instance on package units. If the system is already loaded with the EDDL files, these devices can easily be integrated and commissioned. EDDL files are small and the entire device library only takes up a tiny fraction of the hard disk capacity.

The device revision of the EDDL file shall match the device revision[14] of the device. This is most important. DD (file) revision is not documented because it is not required by the system

5.4 Integration and FAT

At Factory Acceptance Test (FAT) it is common to connect one device of each type, of each protocol (HART, FOUNDATION fieldbus, and PROFIBUS), from each manufacturer. If the EDDL file or internal system files for a particular device have not been loaded on one of the workstations, it will prompt. This is a good way to physically check all EDDL and internal system files are loaded or generated to minimize surprises at site.

5.5 Device Installation

There are no EDDL related activities at device installation.

5.6 Commissioning

A system or handheld needs the correct EDDL file and internal system files to be loaded in order for it to configure and calibrate a device before commissioning. This applies to 4-20 mA/HART, Fieldbus, PROFIBUS, and WirelessHART. Therefore, make sure to have all EDDL and internal system files available and loaded before commission starts.

If a tag in the system database is created for a FF device of a particular revision, but another device revision is shipped to site, that FF device cannot be commissioned on many systems. The configuration for the device tag must be changed in the system database to a device revision matching[15] the actual device before it can be downloaded. For instance, the device manufacturer may have shipped a device of a different device revision than originally stated. Or a device received on one of the package units may be of a different device revision. In a replacement scenario, it is likely the new device has a newer version than the original device being replaced. Depending on the system, the configuration change can be relatively easy. Starting from device profile version 6 (as

[14] for FF the DD (file) revision of the file must be higher than or equal to what is specified in the DD_REV parameter of the device - which is the minimum DD (file) revision required but this has not been a problem so far

[15] Some devices and systems now support 'backwards compatibility' / 'forward compatibility' meaning a newer version device can work with a DD file for an older version device

tested by ITK6), FF devices include a backwards compatibility revision parameter (COMPATIBILITY_REV) which systems can use to determine if the old system database configuration is compatible with the newer device. This new feature will eliminate the need to recreate the tag in the system database with a new device revision. This will make FF device replacement very much easier.

5.7 SAT

During Site Acceptance Test (SAT) optionally test access to different device types from every workstation to make sure the EDDL files and internal system files have been loaded.

5.8 Training for Competency

Training is a critical success factor for any project. Device revision management must be part of the training when a new system is deployed. The device revision management training should be customized for different roles and responsibilities, such as for instrument technicians and system administrators. For instance, instrument technicians responsible for troubleshooting must learn to distinguish a missing EDDL file from a device failure or communication error. System administrators must learn how to keep the system up to date with new types and versions of devices. The training shall also be customized for the systems and portables used at site. The material in this application guide is a good basis for a curriculum.

Appendix 6 Manufacturer ID

All IDs in as used in folder names and documentation are presented in hexadecimal. Humans need not memorize manufacturer IDs, the systems take care of it automatically, displaying manufacturer name in plain text. Note that the manufacturer IDs are not the same for the different protocols.

Table 9 Manufacturer ID of some device manufacturers

Manufacturer	HART	FF
ABB	0005	000320
ABB	0012	
ABB	0016	
ABB	001a	
Berthold Technologies	00a1	00a1ff
Biffi	00b7	424946
Brooks Instrument	000a	000246
CSI	601a	435349
Dresser Valve Division	0065	445644
Dynisco Instruments	0072	44594E
Endress+Hauser	0011	452b48
Fisher Controls	0013	005100
Fisher-Rosemount Performance Technologies	007d	0004D2
Flowserve	0030	464c53
Fuji Electric	0015	000309
Honeywell	0017	48574c
HuaKong Technology	0085	0022B8
Invensys	00a9	385884
Knick	0061	000102
KOSO	00cf	00094b
KROHNE Messtechnik	0045	00012c
K-TEK	0050	000101
Magnetrol	0056	000156
Measurement Technology	0040	0BE0EC
Metso Automation	002f	000e05
Metso Automation	0057	000e05
Mettler Toledo	008e	465255
Micro Motion	001f	000310
Mobrey	003b	000103
M-System	001d	010253
Ohmart	0067	000457
OVAL Corporation	0064	00043E
Pepperl+Fuchs	005d	502b46
PR Electronics	006d	0007d0
R. Stahl	009e	0001F4
Rosemount	0026	001151
Rosemount Analytical	002e	524149
Samson	0042	00e099
Siemens	002a	534147
SMAR	003e	000302
TopWorx	6014	545758
VEGA	0062	56474b
Westlock Controls	004d	574343

Manufacturer	HART	FF
Yamatake	0036	0dfc96
Yokogawa	0037	594543

Appendix 7 Frequently Asked Question (FAQ)

Post your questions on device integration, device revision management, and intelligent device management with the EDDL group on Linkedin:
http://www.linkedin.com/groups?gid=3736433

7.1 Why is there 0x in front of the revision number and manufacturer ID?

0x indicates the number is in hexadecimal format. For instance, 0x17 equals revision 23 in decimal.

7.2 What is the difference between DD and EDDL?

DD is the old name used by the HART Communication Foundation and the Fieldbus Foundation. PROFIBUS International has called it EDDL all along. In 2004 when the EDDL Cooperation Team (ECT) was formed, the common name agreed up on is EDDL. However, DD abbreviation is still often used. Note that EDDL is a collective name covering both the original technology from 1992 as well as the enhanced technology with graphics, advanced menu system, and persistent data storage.

Table 10 Comparison of original EDDL vs. EDDL with enhancements

	Original EDDL	EDDL with Enhancements
Alias	DD, DDL, EDD, DD v4	eEDDL, Enhanced EDDL, DD v5
Year	1992	2006
Logo	EDDL ELECTRONIC DEVICE DESCRIPTION LANGUAGE	EDDL™
Graphics support	No	Yes
Menu support	Limited	Yes
Persistent data support	No	Yes
Standard	IEC 61804-2 Ed 1	IEC 61804-3

Upgrading an old system from traditional DD technology to EDDL technology does not require change of skills. EDDL files are copied and pasted just like HART and Fieldbus traditional DD files are copied today, no software installation has to be learnt.

7.3 Why can't the EDDL file for an older version device be used with the new version device?

The EDDL file for one revision of a device is not used for other revisions of a device because the new device has new features the old device did not have. Therefore every revision of a device type requires its own EDDL file.

It is somewhat like the old printer's driver is not compatible with the new printer. The driver is the way to make the computer compatible with the new printer. The new printer is different from the old printer, that's why a new printer driver is needed.

System software and handheld field communicator are forwards and backwards compatible with devices using the 4-20 mA/HART, FF, PROFIBUS, and WirelessHART digital communication technologies thanks to EDDL technology. That is, an EDDL file is created by the device manufacturer especially for one specific version of a device, the EDDL file enables any system to

work with that version of the device. There is one file for every version of the device. Therefore load the EDDL files for all versions on the system. This way any system can work with any version device, older or newer than the system itself, accessing all features of that device, not just a subset. An EDDL file for one version is not fully compatible with another version device, instead, full backwards and forwards capability is provided by individual EDDL files. That is, EDDL ensures backwards and forwards compatibility between the system and devices by loading the EDDL file matching[16] the device revision used. When a manufacturer creates a new version device, they use EDDL to declare to the system the features of the new device so that old systems can use this new device. Devices keep changing, but software does not change. The EDDL file fills the gap.

There is some level of backwards/forwards compatibility between an old EDDL file and a new device version. Using the old EDDL file created for a prior version of the device, the system can only access the features in the new device described in the old EDDL file. That is, new device features will not be accessible using the old file. In any case, the old file can be sufficient for the system to commission the new device in an urgent replacement scenario. The correct matching EDDL file can be loaded at a later date, enabling full access to all new features in the new device.

7.4 Does "DD revision" have to match "device revision"?

No. This is wrong. Many assume that DD revision of the file must equal device revision of the device, but that's NOT how it works. The device revision of the file shall match the device revision of the device.

7.5 Why do some devices have different EDDL files for different systems?

Some (older) systems do not support conditionals in the EDDL file but most (new) systems do, so device manufacturers need to have two EDDL files. This is not a problem with EDDL technology, it is a limitation in some early systems which don't implement all EDDL features. Testing and registration ensures better support for EDDL in new systems (see appendix 4).

To support older limited systems, device manufacturers have a simplified/non-conditional EDDL-file. This EDDL-file will work on all systems. The device manufacturer may identify this simplified EDDL file by an odd DD (file) revision number (e.g. 0x01). Since many systems do support conditionals, the device manufacturer provides full-featured EDDL-files for these customers. The device manufacturer may identify this full-featured EDDL file by an even DD (file) revision number (e.g. 0x02). That is, the system manufacturer does not modify the DD file, it is the device manufacturer that supplies the simplified file.

That is, since (most) systems don't pay attention to DD (file) revision, device manufacturers sometimes use the DD (file) revision to distinguish between the full-featured and simplified version of the EDDL file.

7.6 Why do some devices appear without graphics?

If the device is not supplied with an EDDL files with graphical enhancements, but only has a traditional DD file, the device manufacturer has not defined any graphics to make the device easy to use, and therefore the display is just a listing of the parameters in the device

[16] Some devices and systems now additionally support 'backwards compatibility' and 'forwards compatibility' between devices and DD files enabling a system to also work with a newer version device using a DD file for an older version device.

Appendix 8 Plant Operations & Maintenance

EDDL plays an important role in simplifying device revision management and system administration tasks during the operations & maintenance phase of the system lifecycle.

8.1 Device Replacement

When replacing an old device with a newer device of a higher revision, make sure to first obtain the EDDL file and internal system files for that device and load it on the system. The device may not be able to be commissioned or fully configured or calibrated without the EDDL file of matching[17] device revision. This applies to 4-20 mA/HART, Fieldbus, PROFIBUS, and WirelessHART. If the system supports backwards compatibility, it can access the features associated with the old version of the device also in a new version of the device using the EDDL file for the old version of the device, but cannot access the new features associated with the new version of the device.

8.2 Device Upgrade

When device firmware is upgraded to take advantage of new feature enhancements, make sure to first have the new EDDL file on hand. That is, find out which device revision the device will become after upgrade, obtain the EDDL file for that device revision, and load it on the system. The device cannot be commission or fully configured without the EDDL file of matching[18] device revision. This applies equally for firmware download on a Fieldbus device as circuit board replacement on a 4-20 mA/HART or PROFIBUS device. Without the DD and internal system file the device cannot be commissioned. If the system supports backwards compatibility, it can access the features associated with the old version of the device also in a new version of the device using the EDDL file for the old version of the device, but cannot access the new features associated with the new version of the device.

8.3 Updated Internal System Files

The system may require additional internal system files for each device type to support the device (device diagnostic alarm management, and possibly proprietary graphics for systems not supporting EDDL enhancements). The files must be obtained from the system manufacturer before a device is upgraded or replaced. Make sure the system supplier has made these files available (see appendix 3). This applies to 4-20 mA/HART, Fieldbus, PROFIBUS, and WirelessHART. Most modern DCS do not have this problem as they are able to auto-generate the internal system files from the standard device files.

8.4 Operating System Upgrade

EDDL files are compressed text, not software, independent of Windows operating system. Therefore there is no need to obtain new EDDL files if the Windows or NET framework will be upgraded, or if service packs or hot fixes will be installed.

[17] Some devices and systems now support '*backwards compatibility*' (Fieldbus terminology) and '*forwards compatibility*' (HART terminology) where newer version devices can be used with DD files created for older version devices

[18] Some devices and systems now support '*backwards compatibility*' (Fieldbus terminology) and '*forwards compatibility*' (HART terminology) where newer version devices can be used with DD files created for older version devices