IEC 62769-4

Edition 2.0 2021-02
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Field device integration (FDI) – Part 4: FDI Packages

IEC 62769-4:2021-02 RLV(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

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

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC online collection - oc.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 62769-4

Edition 2.0 2021-02
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Field device integration (FDI) – Part 4: FDI Packages

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-9391-1

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

FIELD DEVICE INTEGRATION (FDI) –

Part 4: FDI Packages

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

This redline version of the official IEC Standard allows the user to identify the changes made to the previous edition IEC 62769-4:2015. A vertical bar appears in the margin wherever a change has been made. Additions are in green text, deletions are in strikethrough red text.

International Standard IEC 62769-4 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) support for Package Developers to build EDDs targeted for today's EDD bases system under a single development tool;
b) digital signature now includes trusted timestamping for long-term validation of FDI Package;
c) time stamp for device package signature.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
65E/761/FDIS	65E/771/RVD

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

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

A list of all parts in the IEC 62769 series, published under the general title *Field Device Integration (FDI)*, can be found on the IEC website.

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

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

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

INTRODUCTION

~~The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning~~

~~a) method for the supplying and installation of device-specific functionalities, see Patent Family DE10357276;~~

~~b) method and device for accessing a functional module of automation system, see Patent Family EP2182418;~~

~~c) methods and apparatus to reduce memory requirements for process control system software applications, see Patent Family US2013232186;~~

~~d) extensible device object model, see Patent Family US12/893,680.~~

~~IEC takes no position concerning the evidence, validity and scope of this patent right.~~

~~The holders of these patent rights have assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from:~~

~~a) ABB Research Ltd~~
~~Claes Rytoft~~
~~Affolterstrasse 4~~
~~Zurich, 8050~~
~~Switzerland~~

~~b) Phoenix Contact GmbH & Co KG~~
~~Intellectual Property, Licenses & Standards~~
~~Flachsmarktstrasse 8, 32825 Blomberg~~
~~Germany~~

~~c) Fisher Controls International LLC~~
~~John Dilger, Emerson Process Management LLLP~~
~~301 S. 1st Avenue, Marshaltown, Iowa 50158~~
~~USA~~

~~d) Rockwell Automation Technologies, Inc.~~
~~1 Allen-Bradley Drive~~
~~Mayfield Heights, Ohio 44124~~
~~USA~~

~~Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.~~

~~ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.~~

The IEC 62769 series has the general title *Field Device Integration (FDI)* and the following parts:

– Part 1: Overview
– Part 2: FDI Client
– Part 3: FDI Server
– Part 4: FDI Packages
– Part 5: FDI Information Model
– Part 6: FDI Technology Mapping
– Part 7: FDI Communication Devices

- Part 100: Profiles – Generic Protocol Extensions
- Part 101-1: Profiles – Foundation Fieldbus H1
- Part 101-2: Profiles – Foundation Fieldbus HSE
- Part 103-1: Profiles – PROFIBUS
- Part 103-4: Profiles – PROFINET
- Part 109-1: Profiles – HART and WirelessHART
- Part 115-2: Profiles – Protocol-specific Definitions for Modbus RTU
- Part 150-1: Profiles – ISA 100.11a

FIELD DEVICE INTEGRATION (FDI) –

Part 4: FDI Packages

1 Scope

This part of IEC 62769 specifies the FDI Packages. The overall FDI architecture is illustrated in Figure 1. The architectural components that are within the scope of this document have been highlighted in Figure 1.

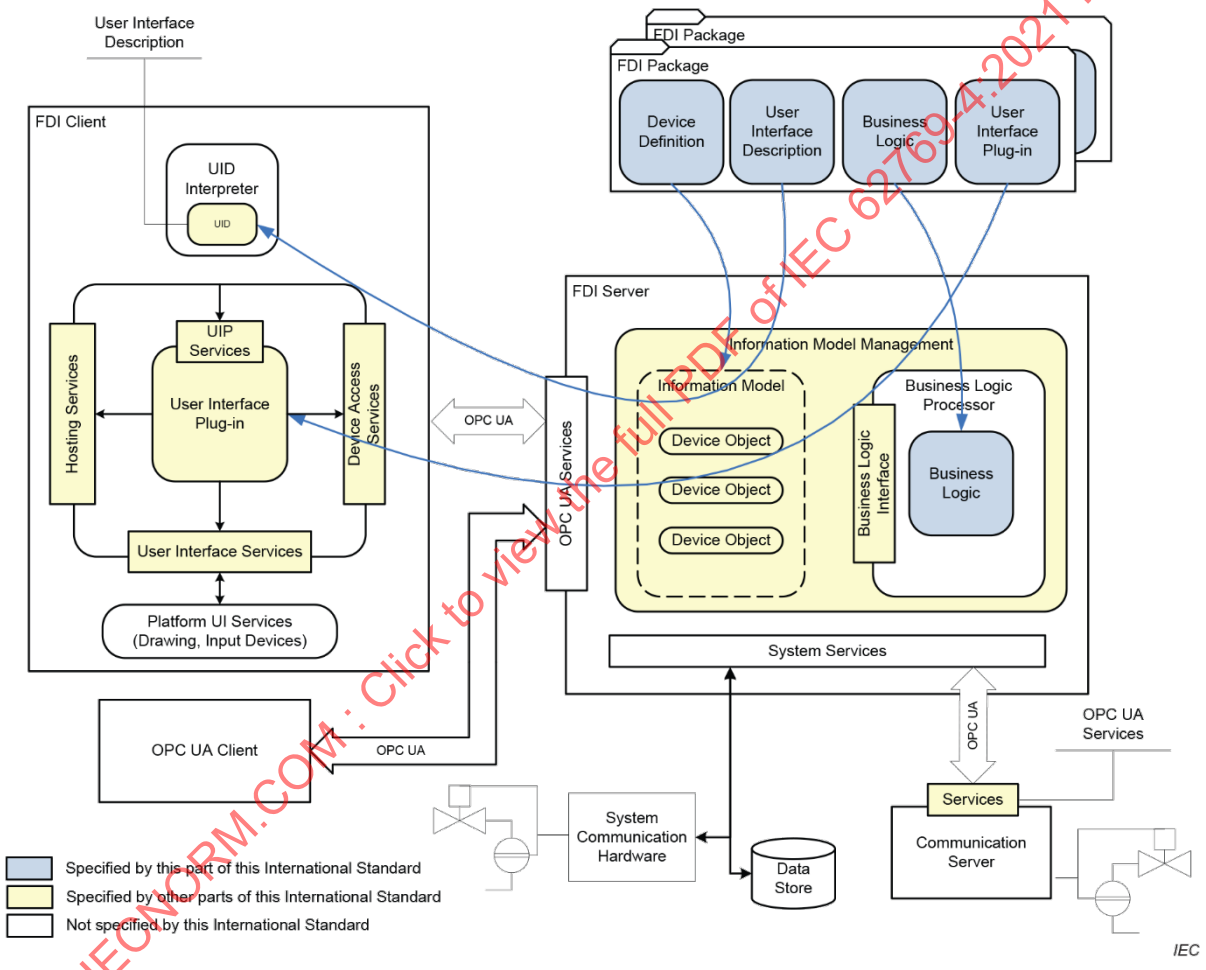

Figure 1 – FDI architecture diagram

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61804 (all parts), *Function blocks (FB) for process control and electronic device description language (EDDL)*

~~IEC 61804-3[1], *Function blocks (FB) for process control and Electronic Device Description Language (EDDL) – Part 3: EDDL syntax and semantics*~~

~~IEC 61804-4:–[2], *Function blocks (FB) for process control and Electronic device description language (EDDL) – Part 4: EDD interpretation*~~

IEC 61804-5:2015, *Function blocks (FB) for process control and electronic device description language (EDDL) – Part 5: EDDL Builtin library*

IEC 62769-1, *Field Device Integration (FDI) – Part 1: Overview*

~~NOTE IEC 62769-1 is technically identical to FDI-2021.~~

~~IEC 62769-5, *Field Device Integration (FDI) – Part 5: FDI Information Model*~~

~~NOTE IEC 62769-5 is technically identical to FDI-2025.~~

IEC 62769-6, *Field Device Integration (FDI) – Part 6: FDI Technology Mapping*

~~NOTE IEC 62769-6 is technically identical to FDI-2026.~~

~~IEC 62769-7, *Field Device Integration (FDI) – Part 7: FDI Communication Devices*~~

~~NOTE IEC 62769-7 is technically identical to FDI-2027.~~

~~ISO/IEC 11578, *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)*~~

ISO/IEC 29500-2:~~2011~~2016, *Information technology – Document description and processing languages – Office Open XML File Formats – Part 2: Open Packaging Conventions*

ISO 639-1, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 32000-1, *Document management – Portable document format – Part 1: PDF 1.7*

Extensible Markup Language (XML) 1.0, W3C Recommendation, available at <http://www.w3.org/TR/REC-xml/>

XML Schema Definition Language (XSD) 1.1, W3C Recommendation, available at <http://www.w3.org/TR/xmlschema11-1/>

ETSI EN 319 132-1, *Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures*

ETSI TS 101 733, *Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)*

FIPS 140-2, *Security Requirements for Cryptographic Modules*

~~[1] To be published.~~

~~[2] To be published.~~

3 Terms, definitions, abbreviated terms and ~~acronyms~~ conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1, ISO/IEC 29500-2, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
Attachment
device and protocol-specific support files that are not directly used to integrate the Device into the system

3.1.2
FDI Communication Package
FDI Package that provides information to integrate an FDI Communication Server to an FDI Server

~~Note 1 to entry: This note applies to the French language only.~~

3.1.3
FDI Device Package
FDI Package that provides one or more device types to an FDI Server

~~Note 1 to entry: This note applies to the French language only.~~

3.1.4
FDI Package Model
description of the structure and elements of an FDI Package

~~Note 1 to entry: This note applies to the French language only.~~

3.1.5
FDI Profile Package
FDI Package that provides information for creating a device type node that can be associated with a class of devices

~~Note 1 to entry: This note applies to the French language only.~~

3.1.6
FDI Registration Authority
entity that has the right and the ability to perform FDI conformance tests on FDI Packages and to issue registration certificate documents

3.1.7
FDI UIP Package
FDI Package that provides one or more UIPs to an FDI Server

~~Note 1 to entry: This note applies to the French language only.~~

~~Note 2 to entry: This note applies to the French language only.~~

3.1.8
Package Catalog
file that describes the contents of an FDI Package

~~Note 1 to entry: This note applies to the French language only.~~

3.1.9
UIP Catalog
file that describes the properties of a UIP

~~Note 1 to entry: This note applies to the French language only.~~

3.1.10
UIP Variant
platform-specific element of a User Interface Plug-in

Note 1 to entry: A UIP is composed of one or more variants. For example, one variant may be optimized for portable devices while another variant is optimized for large-screen devices.

~~Note 2 to entry: This note applies to the French language only.~~

3.2 Abbreviated terms ~~and acronyms~~

For the purposes of this document, the abbreviated terms ~~and acronyms~~ given in IEC 62769-1 as well as the following apply.

CFF	Capabilities File for FOUNDATION Fieldbus
ID	Identifier
IDE	Integrated Development Environment
IM	Information Model
PNG	Portable network graphics
ZIP	Zipper (archive file format)

3.3 Conventions

For the purposes of this document, the conventions given in IEC 62769-1 apply.

4 FDI Package Model

4.1 Overview

The FDI Package Model (see Figure 2) provides all the elements necessary to integrate devices, network components and FDI Communication Servers into a system.

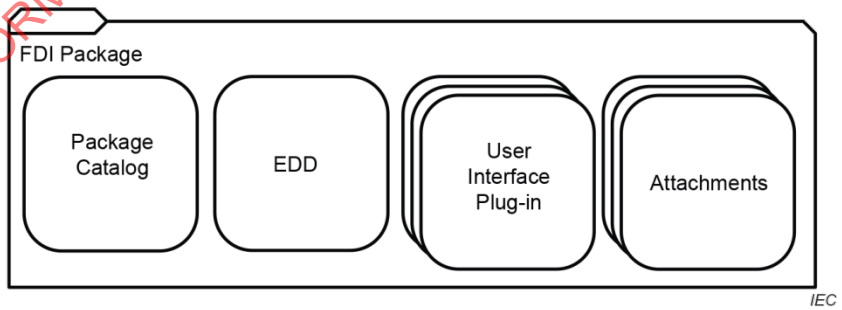

Figure 2 – FDI Package Model

Figure 3 shows the mapping of the FDI Package functional elements, as specified in IEC 62769-1, to the physical elements in an actual FDI Package, as specified in this document.

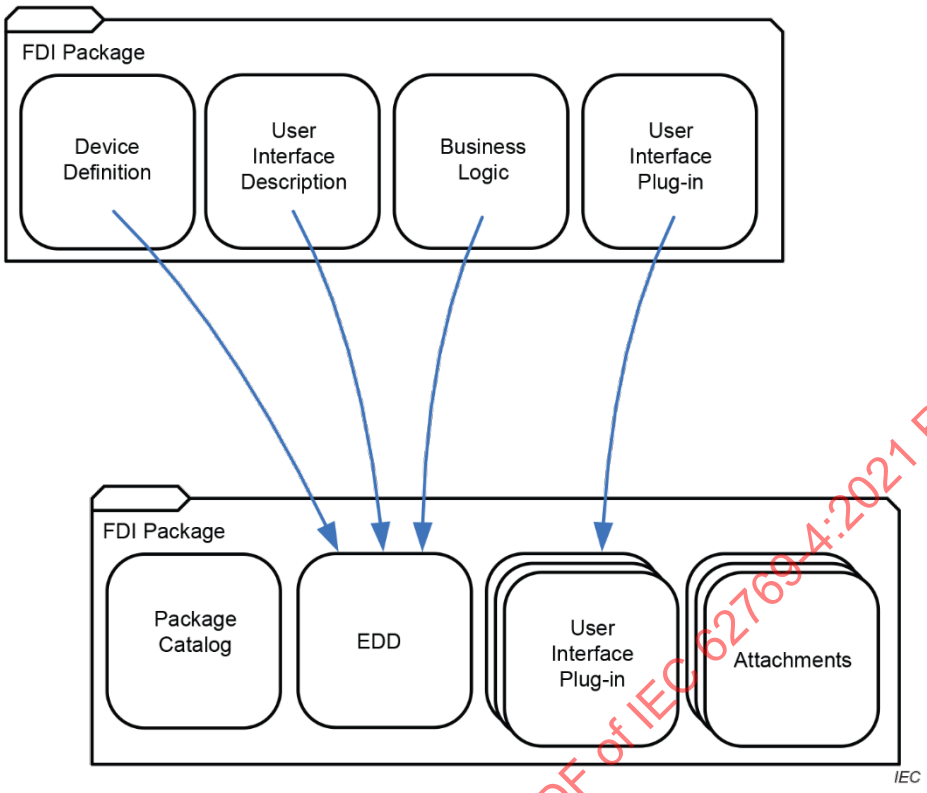

Figure 3 – Architectural mapping

The Electronic Device Description (EDD) corresponds to the Device Definition, the User Interface Description, and the Business Logic functional elements. A set of physical User Interface Plug-ins corresponds to the User Interface Plug-in functional element.

The other physical elements in the FDI Package, the Package Catalog and Attachments, provide support for basic mechanisms such as identification, versioning and deployment, and distribution of manufacturer and/or protocol-specific information about the device and/or the User Interface Plug-ins.

4.2 FDI Package Elements

4.2.1 Package Catalog

The Package Catalog is a required element that provides information about the contents of the FDI Package including, but not limited to, identification and version information, device type information, hardware and input/output device requirements, FDI Technology Version, and protocol-specific characteristics.

The Package Catalog is used by an FDI Server to create a catalog of device types and to create DeviceType Nodes in the Information Model.

4.2.2 Electronic Device Description

The EDD is an element that provides Device Definition, User Interface Descriptions, and Business Logic to an FDI Server.

Subclause 4.3 specifies for which FDI Package types an EDD is required.

To maximize interoperability, the initial setup of a device shall be achievable solely with the User Interface Descriptions, Device Definition and Business Logic that are part of the EDD in the FDI Device Package. The use of User Interface Plug-ins is optional and targeted in particular for the complete setup of complex devices.

4.2.3 User Interface Plug-in

A User Interface Plug-in (UIP) is an element that enables an FDI Client to present a programmed user interface. The FDI Server only stores the UIP from a consumed package; it does not execute or interpret the UIP.

A UIP is referenced from a User Interface Description. Those references are reflected in the Information Model through functional groups (see IEC 62769-5). References to UIPs in the Information Model are logical through a unique reference. The physical structure of the UIP is not at the Information Model level. Physically, a single UIP may consist of one or more UIP Variants, each targeted at a specific platform and run-time environment.

Figure 4 illustrates the logical reference to a UIP in the EDD and the physical structure of that UIP. The EDD of the FDI Package references a globally unique identifier of the UIP. In addition, the Package Catalog also lists the UIPs required by the device type along with the versions of the UIP supported by the package.

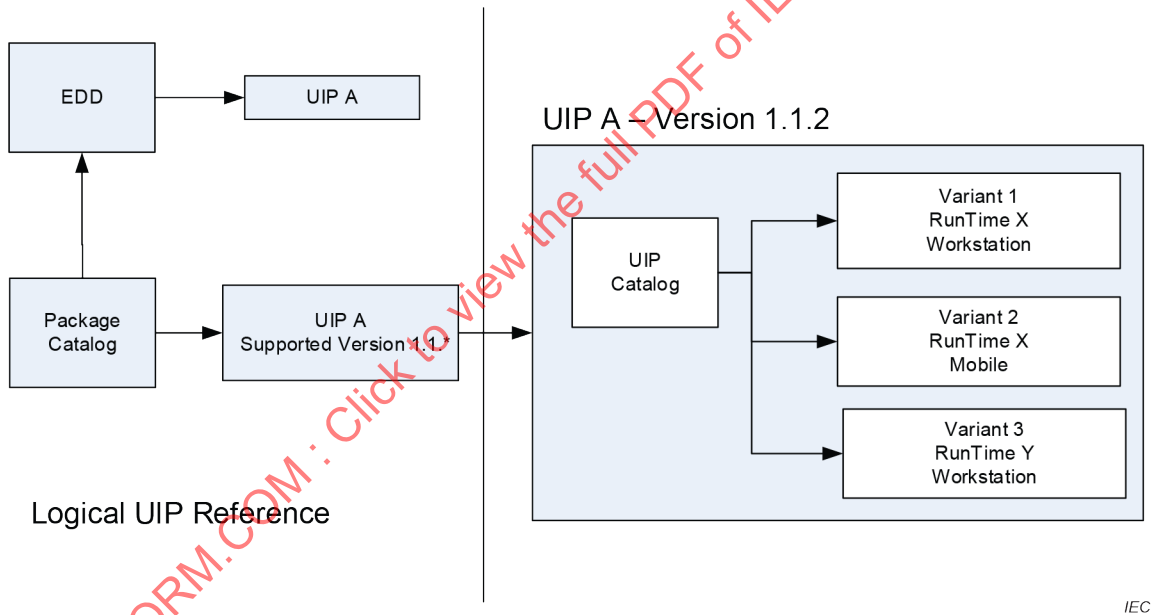

Figure 4 – User Interface Plug-in Reference Model

A UIP can be designed to meet different platform requirements since not all platforms support the same screen sizes and input devices.

When an FDI Client requests a UIP specifying the platform type, the deployment mechanism in the FDI Server looks for a UIP Variant that meets the platform type specified and returns it to the FDI Client.

The target platform defines specific screen resolutions and input devices that shall be supported by the UIP Variant. Available target platforms are described in Table 1.

Table 1 – UIP Platform

Platform	Characteristics
Workstation	A full client typically with a larger display, full keyboard and mouse input.
Mobile	A limited client typically with a smaller display and limited input capabilities.

Detailed platform requirements are technology dependent and specified in IEC 62769-6.

A UIP can be delivered internally (self-contained) or the UIP can be delivered through an FDI UIP Package. An EDD of one package shall not reference any UIP delivered self-contained with another package.

Figure 5 illustrates two FDI Device Packages referencing UIPs from an FDI UIP Package. In this example, the EDD from Device Package A and Device Package B references UIP B delivered by UIP Package C.

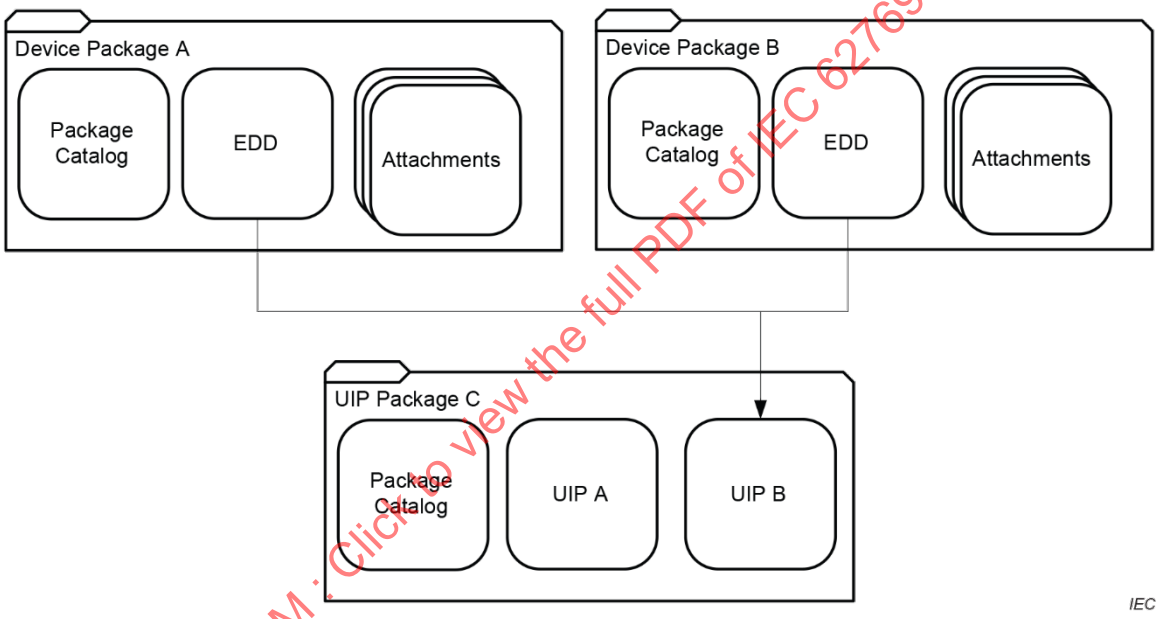

Figure 5 – Multiple FDI Packages referencing a common UIP

4.2.4 Attachment

Attachments provide device and protocol-specific support files, along with other files that are not directly used to integrate the device into the system, for example, product manuals.

The following types of Attachments are defined and reflected in the Package Catalog:

- Protocol-specific files (see 5.3.4.3);
- Documentation (see 5.3.4.2);
- Device pictures (see 5.3.4.1).

Some protocol-specific files are mandatory (see Annex F).

4.3 FDI Package Types

4.3.1 FDI Device Package

The FDI Device Package is intended to provide information about a device to a system. An FDI Device Package describes a single device type. Figure 6 shows the physical structure of an FDI Device Package describing a single device type. Details about how to create a package for a modular device is described in Annex I.

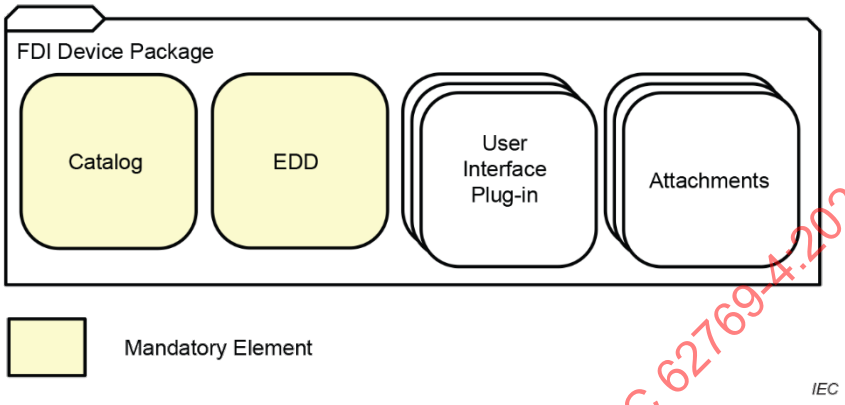

Figure 6 – FDI Device Package

The FDI Device Package shall have a single Package Catalog.

An FDI Package for simple Devices contains one EDD. For modular Devices, multiple EDDs may exist (see Annex I).

The FDI Device Package may include one or more User Interface Plug-ins.

The FDI Device Package may include Attachments.

4.3.2 FDI Communication Package

The FDI Communication Package is intended to provide information about a single FDI Communication Device. The group of FDI Communication Devices splits into two sub-groups.

– FDI Communication Packages for Gateways – Contain everything required to describe all device functions and logic required to bridge between different communication protocol networks, including the bridging algorithms (details are described in IEC 62769-7).

– FDI Communication Packages for FDI Communication Servers – Contain basic information to describe the communication device and to integrate it into the FDI Host but also to create a valid reference to the external FDI Communication Server application. This application shall not be delivered as part of the FDI Communication Package. The FDI Communication Server provides access to single field devices or field device networks (details are described in IEC 62769-7).

  Requirements to the FDI Communication Package content that are specific for FDI Communication Servers are defined in Annex J.

However, representatives of both groups are integrated into FDI Hosts by using FDI Communication Packages. Their physical structure is shown in Figure 7. Binary components necessary to communicate to the communication hardware shall be provided outside the scope of this FDI Communication Package.

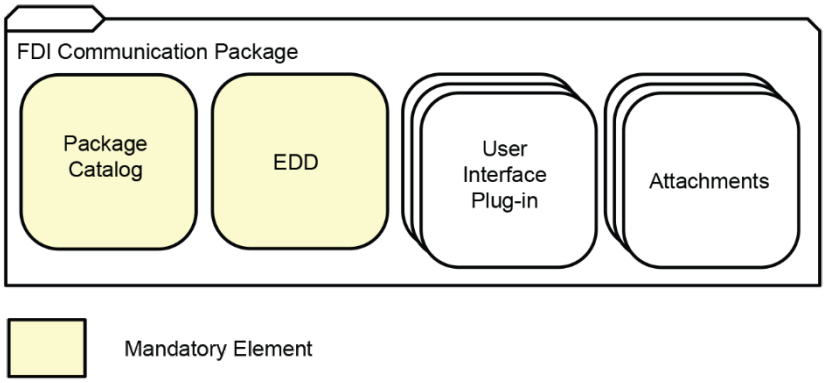

Figure 7 – FDI Communication Package

4.3.3 FDI UIP Package

The FDI UIP Package delivers User Interface Plug-ins to an FDI Server. It is used to distribute one or more UIPs that are intended to be shared by several device types. Its physical structure is illustrated in Figure 8.

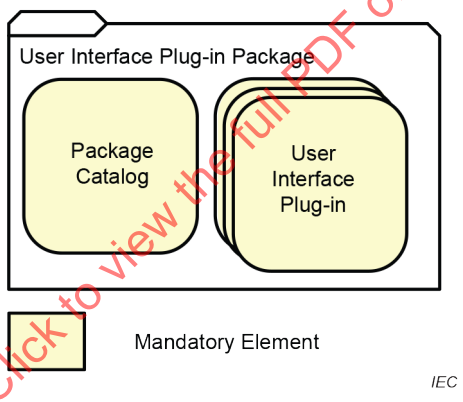

Figure 8 – FDI UIP Package

The FDI UIP Package shall have a Package Catalog and one or more User Interface Plug-ins.

4.3.4 FDI Profile Package

An FDI Profile Package provides information for creating a device type node that can be associated with a class of devices implementing a common set of parameters and functions (described in profiles or profile like definitions created by, for example, communication foundations or similar interest groups).

Conceptually, an FDI Profile Package provides information that is a super class of an FDI Device Package. The FDI Profile Package is represented in Figure 9.

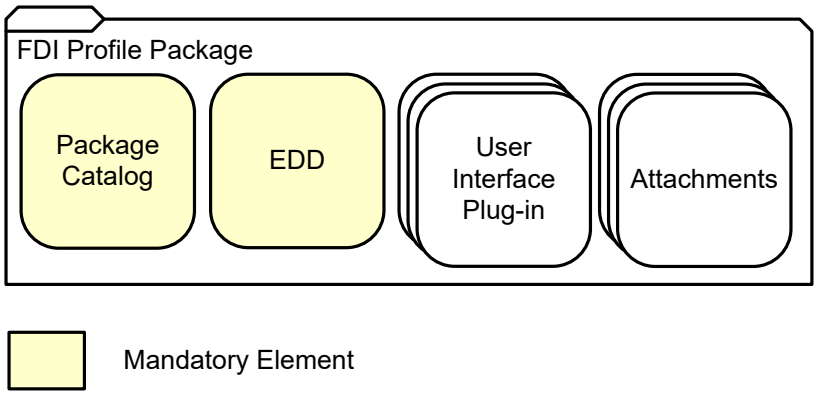

Figure 9 – FDI Profile Package

These packages enable integration of devices without having a specific FDI Device Package. FDI Profile Packages only support standard functionalities as defined by the corresponding communication protocol and application profiles (Annex F).

An FDI Profile Package shall be self-contained. For example, if a Profile B extends the definition of a Profile A, then the FDI Profile Package for the Profile B shall include a complete definition of Profile B, including the elements from Profile A.

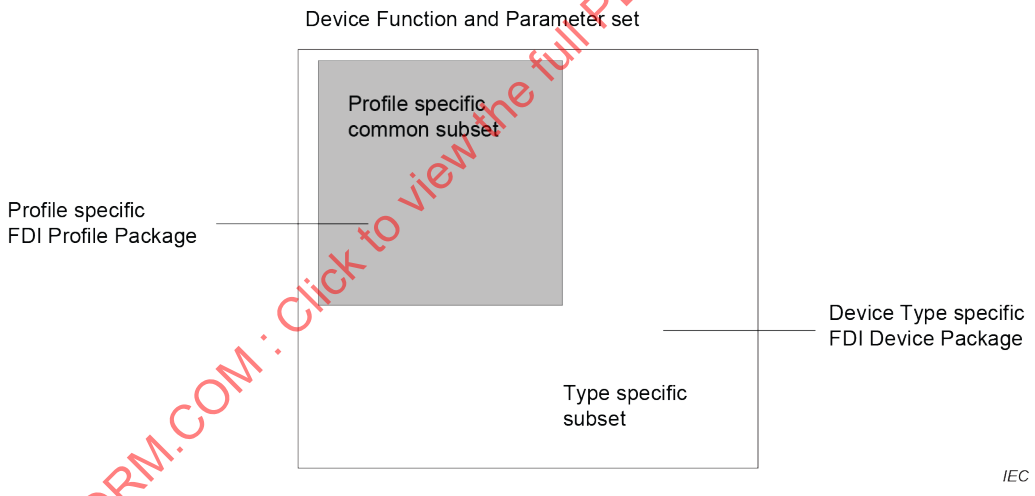

Figure 10 – Device Function and Parameter sets (type- and profile-specific)

Figure 10 represents the entire functionality of a device. Some of the functionality conforms to a specific communication profile, which may be accessed by using an FDI Profile Package. If there is a need to access Device Type specific functionality, in addition to the common functionality, then an FDI Device Package is required.

The detailed description of the communication protocol-specific requirements is not part of this document.

An automatic FDI Package selection implementation (launching FDI Packages matching a connected physical device) typically would search first for available Device Type specific FDI Device Packages. If the specific package is not available, the profile specific FDI Profile Package is loaded (if available). This behavior, however, is system-specific.

5 FDI Package implementation

5.1 Packaging technology

The FDI Package's format complies with the Open Packaging Convention as specified by ISO/IEC 29500-2. The Open Packaging Convention specification is designed to represent a broad range of applications. The technology was designed to provide a mechanism for delivering, deploying, and utilizing a set of logically and physically related components in a flexible, extensible, efficient, and open manner.

Subclause 5.2 clarifies the use of specific elements of the Open Packaging Conventions specification.

The naming conventions for the FDI Package file are described in Annex A. FDI Package creation fundamentals are described in Annex B. An example FDI Device Package implementation is described in Annex D.

NOTE *fdi-cooperation.com* was used when the FDI Cooperation LLC existed. The URL has not been changed to not endanger consistency. The URL does not indicate the current technology ownership and may not point to a valid address.

5.2 Use of Open Packaging Conventions

5.2.1 Unknown parts

Unknown parts are all parts not defined in this document or ISO/IEC 29500-2. Unknown parts may exist as the FDI specification is updated and the legacy FDI Server may not have knowledge of these additional parts. Unknown parts shall be ignored by the FDI Server to enable forward compatibility.

5.2.2 Invalid parts

Invalid parts are those parts that do not conform to the naming guidelines or that do not associate with content types specified in this document or ISO/IEC 29500-2. Invalid parts shall not be used in an FDI Package.

5.2.3 Unknown relationships

Unknown relationships are those relationships that are not defined in this document or ISO/IEC 29500-2. Unknown relationships shall be ignored by an FDI Package consumer.

5.2.4 Interleaving

All parts of the FDI Package shall be laid out using simple ordering. The package producer shall not interleave parts.

5.2.5 Core properties

The core properties are not used in the FDI Package Model and shall be ignored by all FDI Package consumers.

5.2.6 Thumbnails

The thumbnail is an optional part of an FDI Package.

5.2.7 Digital signatures

The use of digital signatures as specified in ISO/IEC 29500-2 is a mandatory part of an FDI Package. For more details, see Clause 7.

5.3 FDI Package Parts

5.3.1 Package Catalog

5.3.1.1 Format

The Package Catalog part is an XML file whose schema is defined in Annex E. An FDI Package shall have only one Package Catalog. The Package Catalog shall be identified by a single package relationship. The root element of the file is Catalog. The structure of a Catalog element is illustrated in Figure 11.

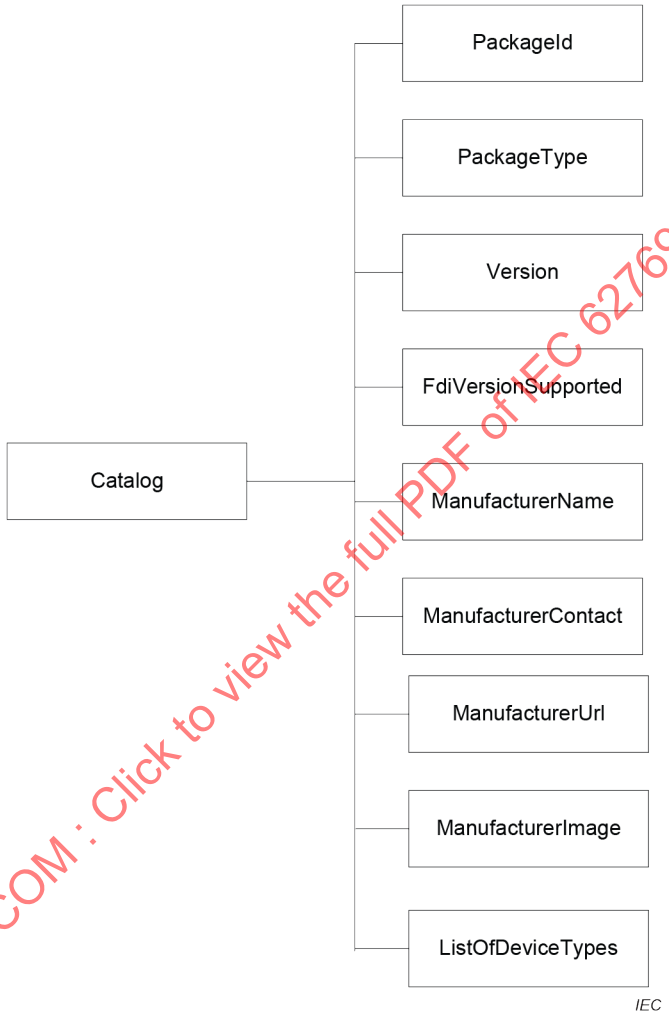

Figure 11 – Catalog Element

The Package Catalog part is described in Table 2.

Table 2 – Package Catalog Part

Part	Content
Content Type	application/vnd.fdi.package.catalog+xml
Root Namespace	http://fdi-cooperation.com/2010/package-catalog
Source Relationship	http://fdi-cooperation.com/2010/relationships/package-catalog
Filename	catalog.xml

5.3.1.2 Content

The schema for the Package Catalog is described in Annex E.

5.3.2 Electronic Device Description

5.3.2.1 Format

The EDD is an encoded file using the Electronic Device Description Language (EDDL) in accordance with 5.3.2.2.

The format of the EDD part is described in Table 3.

Table 3 – EDD part

Part	Content
Content Type	application/vnd.fdi.package.edd
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/edd
Filename	Not specified

5.3.2.2 Content

The EDDL is specified in IEC 61804. FDI specific rules for EDDs are described in Annex K.

5.3.3 User Interface Plug-in

5.3.3.1 Format

An FDI Package may contain one or more User Interface Plug-in parts. All User Interface Plug-in parts shall be identified by a package relationship.

The User Interface Plug-In represents a container element for UIP Variants (see 5.3.3.2.2.2), which contain the physical representation of a UIP (different variants for different platforms) that is consumed and executed by an FDI Client. A User Interface Plug-in shall at least provide one UIP Variant.

The variants are packaged into a single User Interface Plug-in using Open Packaging Convention as specified in 5.3.3.2. The User Interface Plug-in is not directly consumed by an FDI Server. The User Interface Plug-in is an optional part of different FDI Package types as specified in 4.3. The User Interface Plug-in is illustrated in Figure 12.

The UIP version information as well as the update and upgrade behavior shall follow the rules and concepts described in IEC 61804-1.

The default locale for UIPs and all contained variants is English (US). Optional language support is allowed according to market needs.

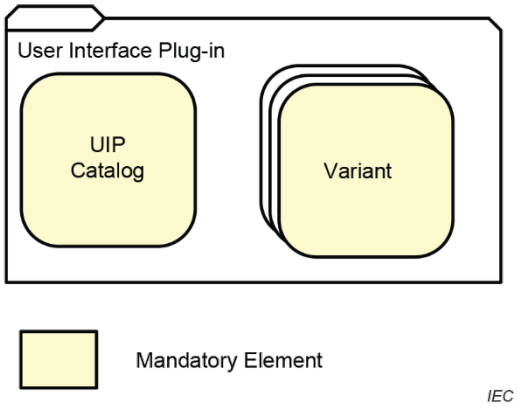

Figure 12 – User Interface Plug-in

The format for the User Interface Plug-in part is described in Table 4.

Table 4 – User Interface Plug-in part

Part	Content
Content Type	application/vnd.fdi.package.uip
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip
Filename	extension shall be .uip

5.3.3.2 Content

5.3.3.2.1 Use of Open Packaging Convention

5.3.3.2.1.1 Core

Core Properties are not used and shall be ignored by all FDI UIP Package consumers.

5.3.3.2.1.2 Thumbnails

Thumbnails are not used and shall be ignored by all FDI UIP Package consumers.

5.3.3.2.1.3 Digital Signatures

Digital signatures might be necessary for UIP Variants (see 5.3.3.2.2.2) depending on the implementation technology. Implementation technology details and necessary mechanisms to be implemented are described in IEC 62769-6.

The User Interface Plug-in as such however shall not be signed or applied signatures shall be ignored by the consuming FDI component.

5.3.3.2.2 User Interface Plug-in Parts

5.3.3.2.2.1 UIP Catalog

5.3.3.2.2.1.1 Format

The UIP Catalog is an XML file describing the properties of a UIP that are necessary to create the Information Model nodes in the FDI Server side and for deployment (FDI Server to FDI Client).

The UIP Catalog also hosts information about the physical starting element that has to be called/executed from a UIP Variant on the FDI Client. The format and type of this starting element are technology specific and therefore defined in IEC 62769-6.

The UIP Catalog shall be identified by a single package relationship.

The UIP Catalog part is an XML file whose schema is defined in Annex E. A User Interface Plug-in shall have only one UIP Catalog. The root element of the file is UipCatalog. The structure of the UipCatalog element is illustrated in Figure 13.

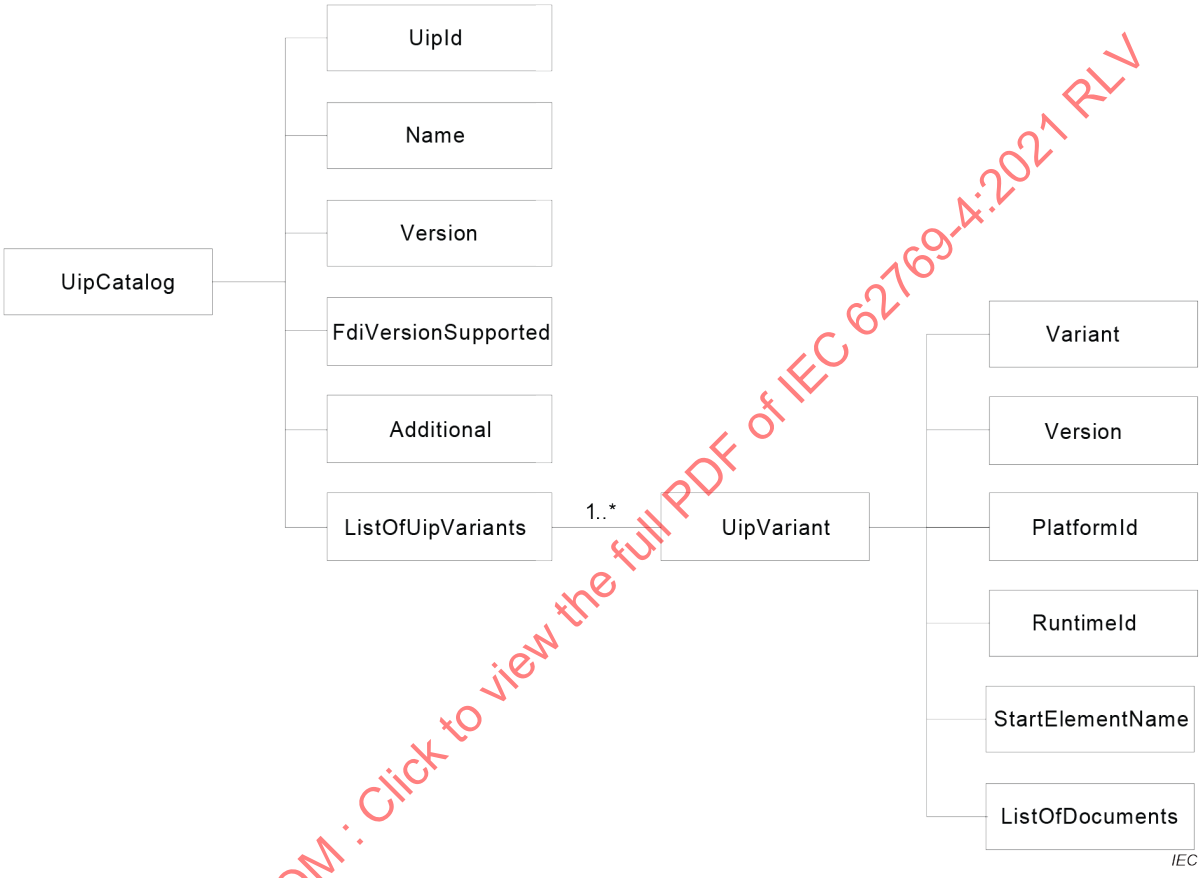

Figure 13 – UIP Catalog

The UIP Catalog part is described in Table 5.

Table 5 – UIP Catalog Part

Part	Content
Content Type	application/vnd.fdi.package.uip.catalog+xml
Root Namespace	http://fdi-cooperation.com/2010/uip-catalog
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip-catalog
Filename	uipcatalog.xml

5.3.3.2.2.1.2 Content

The schema for the UIP Catalog is described in Annex E.

5.3.3.2.2.2 UIP Variant

5.3.3.2.2.2.1 Format

A UIP Variant can be provided to support different versions of the base technology. It is a container for all physical files that represent the UIP Variant, independently of their type and number. The UIP Variant shall use the ZIP file format (compressed archive, application/zip media type) as bundling technology.

Use of the ZIP specification for bundling variants shall be constrained to the requirements in ISO/IEC 29500-2:~~2011~~2016, Annex C.

The UIP Variant part is described in Table 6.

Table 6 – UIP Variant Part

Part	Content
Content Type	application/zip
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip-variant
Filename	Not specified

5.3.3.2.2.2.2 Content

The content of the UIP Variant is specified in IEC 62769-6.

5.3.4 Attachments

5.3.4.1 Image

An FDI Package image attachment is a computer icon that represents the device type. Multiple images and resolutions of the device type are supported. Each image shall be PNG format and restricted to the following pixel resolutions: 256 × 256, 64 × 64, 32 × 32 or 16 × 16. The image part is described in Table 7.

If the Device Package is supposed to support handheld devices the provided image shall be available in a resolution of 16 pixels × 16 pixels.

Table 7 – Image Part

Part	Content
Content Type	image/png
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-image
Filename	Not specified

5.3.4.2 Documentation

Documentation parts include documentation such as product manuals and data sheets and shall be encoded in PDF or plain text format. The documentation Attachment is described in Table 8.

Table 8 – Documentation Part

Part	Content
Content Type	Application/pdf (specified in ISO 32000-1) Text/plain
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-documentation
Filename	Not specified

5.3.4.3 Protocol Support File

The protocol support file part is a file not defined by this document but necessary to integrate the product into a system. Those files usually extend the integration mechanism provided in the basic part of the FDI Device Package (EDD, UIPs) by protocol-specific means.

Those files shall be neither executables nor binary files of any kind. In addition, those files shall not replace any mandatory elements of an FDI Device Package.

A list of protocol support files for the different communication protocols are defined in Annex F. The protocol-specific support file part is described in Table 9.

Table 9 – Protocol Support File Part

Part	Content
Content Type	Not specified here
Root Namespace	Not specified here
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Filename	Not specified here

5.3.4.4 FDI Registration Certificate

5.3.4.4.1 Format

The FDI Registration certificate is a single XML document per FDI Package, which can be present (optional FDI Package Element). In the scope of an FDI Package, it is a Package Element of the type Attachment.

Table 10 – FDI Registration Certificate Part

Part	Content
Content Type	application/vnd.fdi.package.registrationCert+xml
Root Namespace	Not specified here
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-registrationCert
Filename	RegistrationCert.xml

5.3.4.4.2 Content

The FDI Registration Certificate is machine readable and is represented in XML Format and has the content and structure in accordance with Figure 14. The FDI Registration Certificate may be signed with a detached signature within the same document.

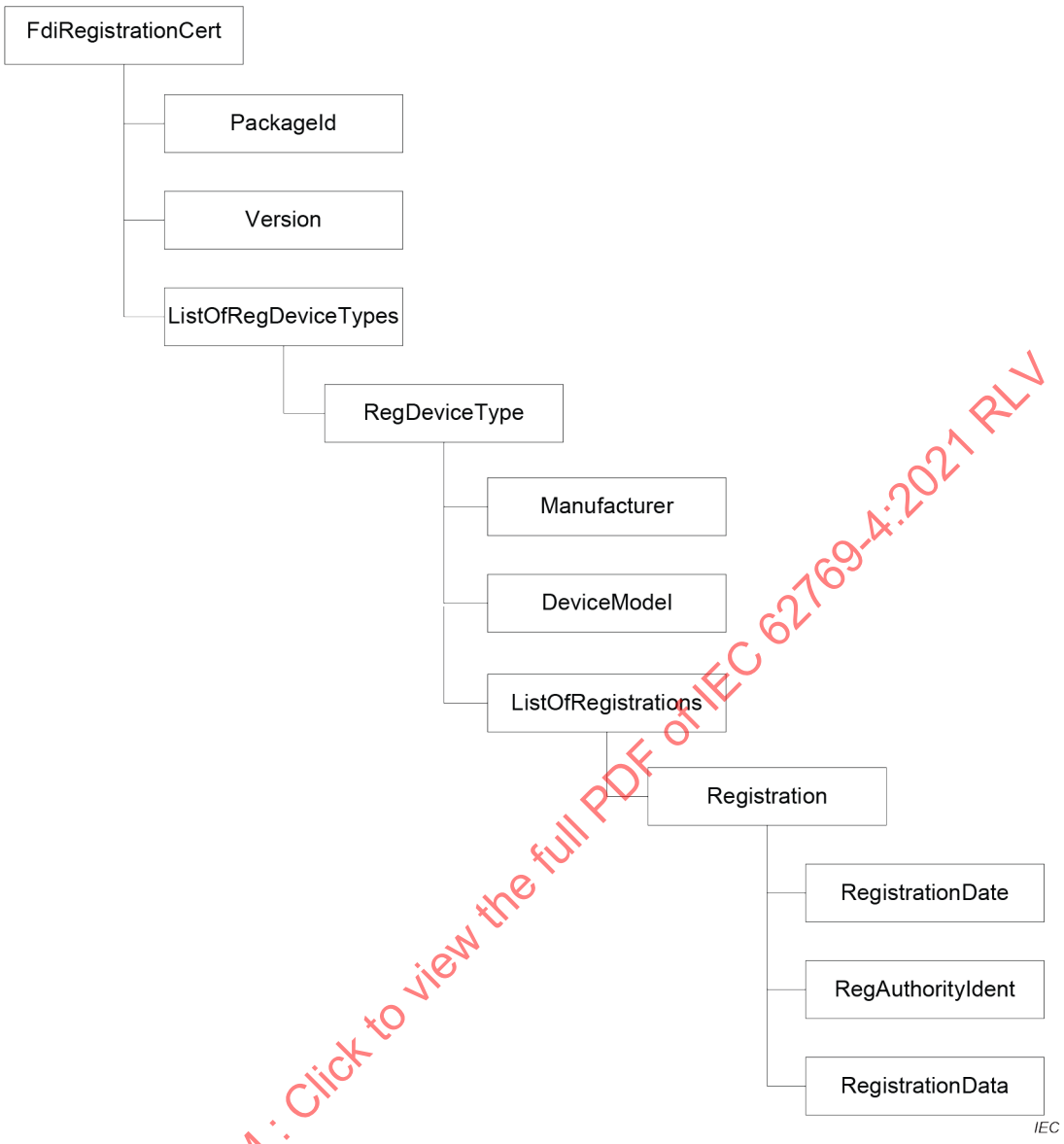

Figure 14 – FDI Registration Certificate

The schema for the FDI Registration Certificate is described in Annex E.

6 FDI Package Versioning

6.1 Version scheme

FDI elements use a major.minor.revision numeric versioning scheme for package and package elements. The initial version shall be 1.0.0.

Incompatible changes are indicated by incrementing the major number. Functional changes that still maintain compatibility to the major version are indicated by incrementing the minor number. Non-functional changes, such as editorial changes are indicated by incrementing the revision number. Rules for incrementing the version number are described in 6.3.

Examples of how to deploy different FDI Package types are described in Annex C.

6.2 Versioned elements

An FDI Package contains the version information that is described in Table 11.

Table 11 – Versioned Elements

Scope	Version Location	Schema Reference
FDI Package (see 5.3)	Package Catalog (see 5.3.1)	See Annex E, Version element of the Package complex type
UIP (see 5.3.3)	UIP Catalog (see 5.3.3.2.2.1)	See Annex E, Version element of the Uip complex type
UIP Variant (See 5.3.3.2.2.2)	UIP Catalog (see 5.3.3.2.2.1)	See Annex E, Version element of the UipVariant complex type

The following parts of the FDI Package have version mechanisms that are outside the scope of the FDI.

- EDD (see 5.3.2)
  - DD_REVISION and DEVICE_REVISION in accordance with IEC 61804-3
- Attachments (see 5.3.4)
  - Image (see 5.3.4.1)
  - Documentation (see 5.3.4.2)
  - Protocol Support File (see 5.3.4.3 and Annex F)

6.3 Version Hierarchy

The version change of FDI Package elements can influence the version of overlaying elements. Figure 15 illustrates those influences. The orange boxes show elements that are versioned using the FDI specific version mechanisms described in IEC 62769-1. The green boxes indicate elements that have version mechanisms that are not described by FDI. The white boxes describe elements that are not explicitly versioned.

All elements in the tree structure shown below influence the version of the parent elements in accordance with Table 12. Their version change causes a version change of the overlaying element.

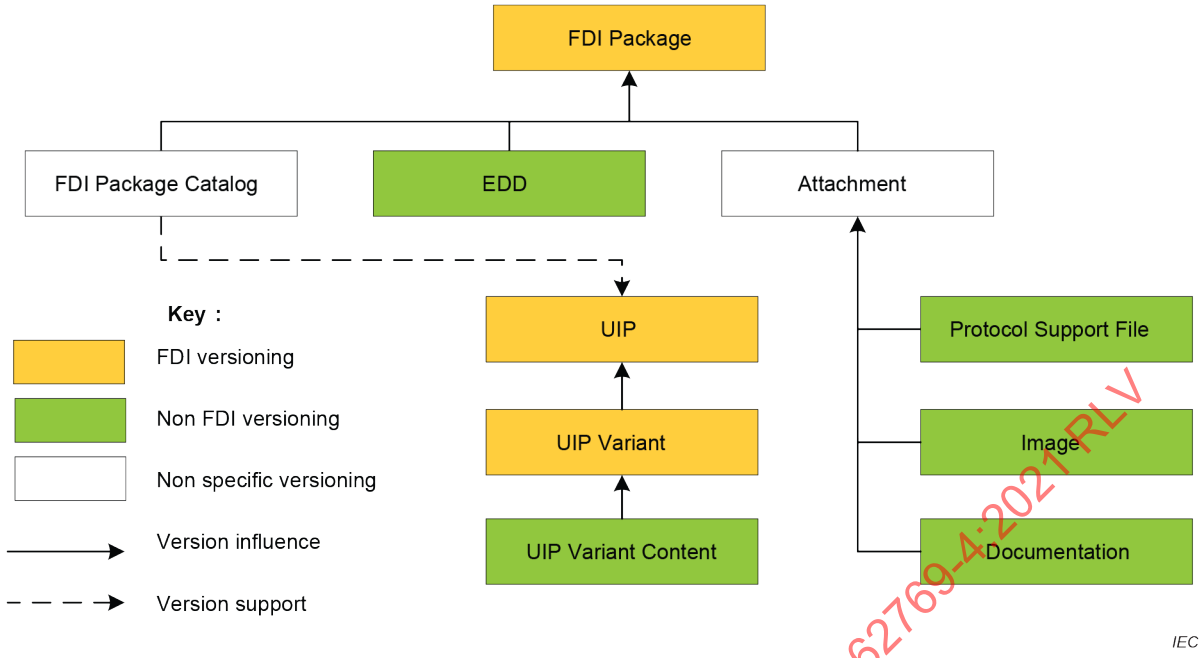

Figure 15 – Version Hierarchy

Table 12 describes a selection of possible changes to the different FDI Package elements and their direct influence on the FDI Package version. At a minimum, successive FDI Package versions shall increment the revision number.

Table 12 – Influence on FDI Package Version

Element	Version Level			FDI Package Type	
	Major	Minor	Revision	Device/ Profile/ Comm Package	UIP Package
Package Catalog	n/a	Modifying existing compatibility references. Change to ListOfSupportedUips (see Annex E). Only addition of wildcards is allowed for modifications to existing VersionSupported (see Annex E)	Editorial	x	x
EDD	Increment Device Revision	Increment DD_REVISION with same DEVICE_REVISION	n/a	x	
Attachments	Changes to Protocol Support Files	Changes to Protocol Support Files	Changes to Documentation, Images, Protocol Support Files	x	
UIP	Incompatible changes	Functional enhancement	Bug fix	x	x
	Technology driven	Technology driven	Technology driven		

Annex G describes typical FDI Package life cycle use cases for a better understanding of the version hierarchy and the versioning concept.

6.4 UIP Compatibility

UIPs are delivered in an FDI Device Package, FDI Communication Package, FDI Profile Package or an FDI UIP Package. A UIP delivered in an FDI UIP Package shall not be delivered in any other package type. Therefore, an FDI Package may not be self-contained.

FDI Device Packages, FDI Communication Packages and FDI Profile Packages do not directly reference UIPs. Instead, the Package Catalog defines the SupportedUip (see Annex E) to define compatible UIPs. More than one installed UIP may be compatible with a given FDI Package version. This permits the FDI Package creators to provide bug fixes and functional enhancements to the FDI Package without the need to provide a major release of an FDI Package.

The version support described in the Package Catalog is defined as follows:

- The major release number shall be specified explicitly as a single number.
- The minor release number and revision number shall be specified explicitly as a single number or as a wildcard ('*'). If the minor release number is specified as a wildcard, then the revision number shall also be specified as a wildcard.

NOTE Examples of valid compatible version numbers are 1.3.1, 1.3.*, 1.*.*.

The use of a wildcard indicates that all minor numbers and/or revision numbers are compatible with the FDI Package. If the SupportedUip has been defined by using wildcards and there is more than one version of the UIP available, the FDI Server shall transfer the latest version to the FDI Client. System-specific implementations that allow coexistence of minor release or revisions releases regardless of version support wildcards are outside the scope of this document.

Figure 16 illustrates an example of selecting the most recent compatible UIP.

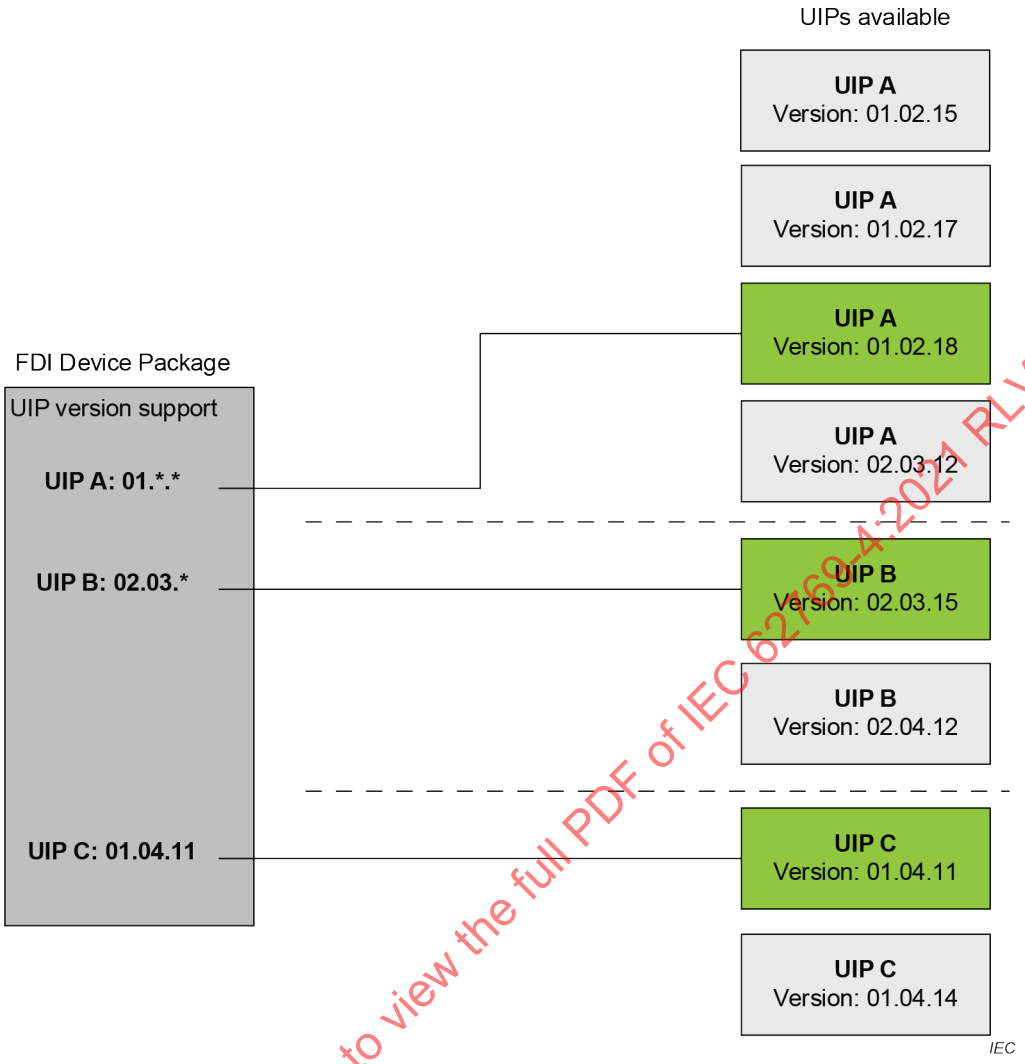

Figure 16 – UIP Version Support concept

7 Digital Signatures and Registration Certificates

7.1 Signed Elements and Certification documents

The FDI Package signing policy defines that the entire FDI Package shall be signed by the FDI Package originator (see 7.3). FDI Packages that have been registered by an FDI Registration Authority should contain an FDI Registration Certificate (special attachment, see 5.3.4.4), which shall be signed by an FDI Registration Authority (see 7.3). FDI Packages are registered if they are successfully conformance tested (see B.2.4).

Registered FDI Packages should carry one or more digitally signed FDI Registration Certificates to:

– indicate that the FDI Packages has been registered by an official FDI registration authority;

– indicate that the files in the package, which had been central for registration of the package, have not been altered after submission for registration.

FDI Packages as a whole (the surrounding entity covering all elements) shall be signed to:

– identify the originator (the signer) of the FDI Package;
– verify that the signed FDI Package hasn't been altered after the signature was applied.

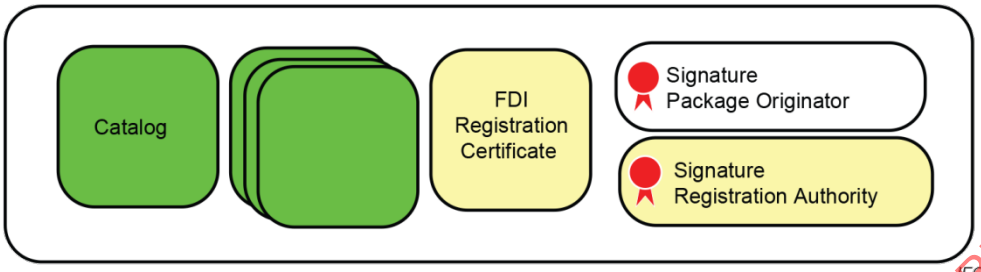

Figure 17 – FDI Package signing

The FDI Package originator first sends a package to the FDI Registration Authority. They perform defined conformance tests and additional tests according to their specific rules and test descriptions or agreements.

After successful testing, those Registration Authorities might issue an FDI Registration Certificate, which shall be signed by the individual FDI Registration Authority. The FDI Registration Certificate may contain more information about the registration, for example a hash on the parts of the FDI Package which had been covered by the conformance tests. The FDI Registration Certificate is incorporated into the FDI Package for release. After adding the FDI Registration Certificate to the FDI Package content, the package is again signed and afterwards released/published by the originator.

The FDI Package originator therefore takes over the responsibility that the FDI Registration Certificate is trustable by signing the entire FDI Package he releases.

7.2 Signing mechanism

~~The FDI Package shall be signed according to the mechanism defined in ISO/IEC 29500-2.~~

All signatures within the FDI Package shall be made in accordance with the mechanism defined in ISO/IEC 29500-2. In addition to ISO/IEC 29500-2, the signature shall fulfill the following requirements:

– The information needed to validate the signature shall be part of the digital signature, i.e. the KeyInfo element specified in XML Signature Syntax and Processing is mandatory.
– Certificates used for signing shall be rooted in a Certificate Authority which is included in the trusted CAs of the Microsoft Windows Certificate Store[3].
– The algorithms used in creation of the signature (for hashing and encryption/decryption) shall be from the list of NIST recommended algorithms in FIPS 140-2, Annex A (NIST).
– The signature shall include a trusted timestamp in compliance with XAdES (XML Advanced Electronic Signatures – ETSI EN 319 132-1).
– Any signature shall include a CommitmentTypeIndication in accordance with ETSI TS 101 733. The used commitment types are specified in 7.3.

[3] Microsoft Windows Certificate Store is the trade name of a product supplied by Microsoft®. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

7.3 FDI Package Originator, FDI Registration Authority

The FDI Package Originator and the FDI Registration Authority have the following responsibilities:

– An FDI Package originator officially publishes an FDI Package and signs it to ensure the integrity of the FDI Package. The FDI Package can be created by a device vendor or a software solution provider. The publisher of an FDI Package may be a different person. The commitment type is ProofOfOrigin.

– An FDI Registration Authority has the right and the ability to perform FDI conformance tests on FDI Packages and to issue FDI Registration Certificates, Typically interest groups representing an FDI supported communication protocol or their authorized partners. The commitment type is ProofOfApproval.

7.4 FDI Host behavior

An FDI host system shall display a warning message when the FDI Package import procedure recognizes that:

– a digital signature on the package is not present or does not include all entities (files) inside the package,

– the digital signature as such is not trustable,

– the signature is broken, which indicates that the package has been modified after signing.

Additional security measures to be taken, if the warning message has been displayed, are in the responsibility of the FDI host system.

An FDI host system should display an information message showing which parts of the ones having gone into the registration have been changed when the FDI Package import procedure recognizes that:

– the unique identifier (PackageID) and the version (as defined in Annex E) of the FDI Package does not match the same information given as a part of the FDI Registration Certificate file;

– there is no FDI Registration Certificate present in the FDI Package;

– the included FDI Registration Certificate is not signed, the signature is not trustable, or the signature is broken.

~~An FDI host system can check the signature and certification status by reading the FDI Registration Certificate. A missing FDI Registration Certificate or a broken signature on the same however shall not stop or prevent the import of this FDI Package into the FDI host system and shall not limit the functionality of the same.~~

An FDI host system can check the signature and certification status by reading the FDI Registration Certificate. A host shall provide a configuration, which allows to import an FDI Package, which does not include an FDI Registration Certificate. The functionality of this FDI Package shall not be limited.

Annex A
(normative)

File name conventions

A.1 Identification

Identification naming conventions shall be utilized to provide a unique way of identifying complete FDI Packages as well as elements of FDI Packages. Adherence to the identification rules will promote interoperability across systems. However, the names themselves shall not be the only mechanism for deployment.

Traditionally, machine-readable naming conventions have been utilized to uniquely link a file to a specific device and facilitate the import and use of device interface files, such as an EDD, making it difficult for users to determine if the required file was available in the file system. Because an FDI Package is the visible element to the user, a human-readable format is preferred over that of a machine-readable version.

A.2 FDI Package filename convention

Individual FDI Packages shall be identified by unique file names, which shall consist of the manufacture, the model or type, the revision and the protocol supported. ~~Due~~ Owing to filename persistence problems, the file name shall not be the only means to identify an FDI Package. Secure means of identifying an FDI Package shall be utilized to prevent inappropriate mixing of device to FDI Packages.

The FDI Package shall use the following naming convention

<manufacturer>.<description>.<major>.<minor>.<revision>.<protocol>.fdix

Each element of the filename is described in Table A.1.

The following rules for naming an FDI Package shall apply:

- All FDI Packages shall use the .fdix extension.
- Maximum name length including .fdix extension: 128 characters.
- The file name shall not include a space.

NOTE The maximum file name length is limited to 128 characters to reduce the probability of a path exceeding the maximum value.

Table A.1 – FDI Package Naming Convention

Filename component	Description
manufacturer	String representation of the manufacturer name of the device. The manufacturer shall not include a period.
description	A brief description of the package. The description shall not include a period. For an FDI Device Package, a string representation of the device type name.
major	Two-character numerical representation of the major release of the FDI Package.
minor	Two-character numerical representation of the minor release of the FDI Package.
revision	Two-character numerical representation of the revision of the FDI Package.
protocol	The communication profile family names are defined in the communication profiles (IEC 62769-1xx-x).

Annex B
(informative)

FDI Package Creation

B.1 General

Annex B describes the fundamentals of a possible FDI Package creation process by using standardized development tools and components available.

B.2 Tools and Components

B.2.1 Overview

FDI Packages consist of several different components – which also might have relationships and dependencies to each other – that are developed by using different implementation technologies and that follow different standards. This complex structure of FDI Packages requires a tool support to make an easy and economic development and maintenance possible.

B.2.2 FDI Reference Implementation/Common EDD Engine

The FDI Reference Implementation, including a common EDD Engine, ensures that a common implementation is used for an FDI Package implementation and test that has a defined behavior. The FDI Reference Implementation is part of both tools mentioned below.

B.2.3 FDI Package IDE

The FDI Package IDE provides everything that is necessary to manage development projects for the different kind of FDI Packages, the development of the descriptive parts, the linking of all remaining package parts, but also to do the actual packaging of the package.

B.2.4 FDI Device Package Conformance Test Tool

A developed FDI Package and especially FDI Device Packages are tested to prove the conformance of the implementation to the FDI Specification. This ensures interoperability. The conformance test is done by using the FDI Device Package Conformance Test Tool that executes defined test cases with the FDI Reference Implementation and the developed FDI Package.

B.3 Development

B.3.1 FDI Package core development

The FDI Package IDE allows the creation of a development project for the FDI Package, including the project type and version information. This development project can be used during the complete life cycle of the FDI Package. A development project wizard might be available to speed up the project creation process.

An editor component coming with the FDI Package IDE is available to implement the descriptive part (EDD) of the FDI Package. Features such as syntax checks, code folding, auto complete and wizards for complex constructs are available to support an efficient and safe development.

The FDI Package (depending on the FDI Package type) consists of several FDI Package parts. A project management component is also part of the FDI Package IDE, which allows the linkage of external FDI Package parts to a specific FDI Package development project. As soon as the developer has linked all FDI Package parts to the development project (and has also implemented EDD relationships if necessary), the FDI Package can be bound and packaged.

The FDI Package IDE generates parts of the Package Catalog depending on linked FDI Package parts, development project information and EDD source code. Some parts of the FDI Package, however, cannot be developed by using the FDI Package IDE. Examples are given ~~below~~ in B.3.2 and B.3.3 and shown in Figure B.1.

B.3.2 User Interface Plug-in development

User Interface Plug-ins are developed using well known implementation technologies (for example, Microsoft.NET[4]). There are several powerful development tools available to use those implementation technologies (for example, Microsoft Visual Studio®[5]). Those tools should be used to implement the User Interface-Plug-ins needed. The ready implemented Plug-ins can then be imported into the FDI Package IDE development project to be referenced in the EDD and to be packaged into the FDI Package for release, but also to be tested under runtime conditions in conjunction with the descriptive part of the FDI Package. See also Figure B.1.

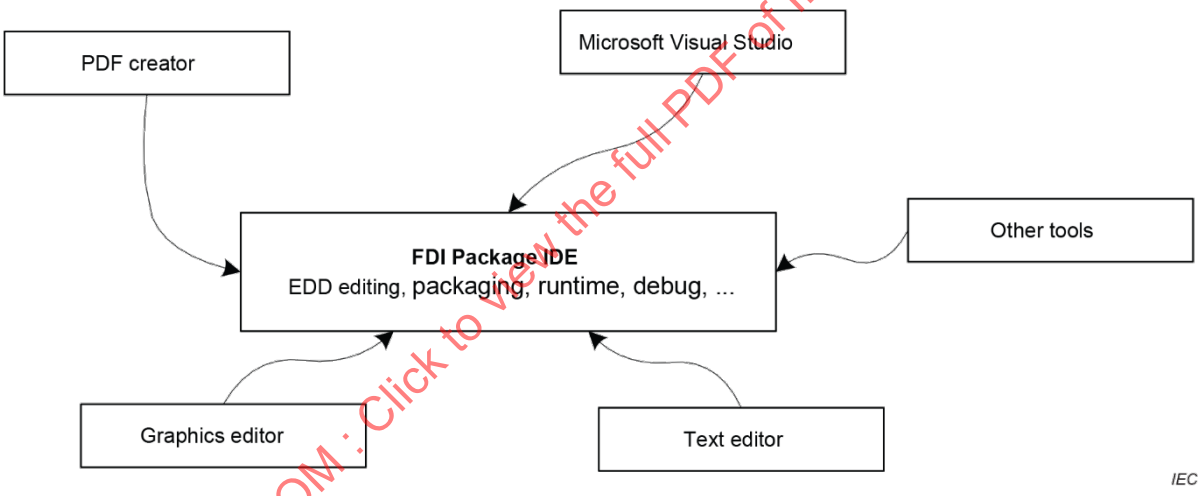

Figure B.1 – Tools used for FDI Package development

B.3.3 FDI Package Attachment development

There are different kinds of Attachments that can be incorporated into FDI Packages. ~~Due~~ Owing to the variety of possible and necessary Attachments, appropriate development tools depending on the type of Attachment shall be used to implement/create those. The Attachments can then be imported into the FDI Package IDE development project to be packaged into the FDI Package for release. See also Figure B.1.

---

4 Microsoft.NET is the trade name of a product supplied by Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

5 Microsoft Visual Studio is the trade name of a product supplied by Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

B.3.4 FDI Package binding and packaging

The last step of the FDI Package development is the packaging as such. In this development step, all developed and related parts of the FDI Package are packed according to this specification. Checks are performed to ensure consistency. The output can then be used for conformance testing and in systems.

B.3.5 Conformance Test

The use of the FDI Package Conformance Test tool which is also part of the FDI Package IDE to ensure the conformance of the FDI Package with the FDI specification marks the last step before releasing the product. The Conformance Test tool uses specified test cases with an FDI runtime engine to check the single features of the FDI that have or may be implemented into an FDI product (an FDI Package in this case).

Annex C
(informative)

FDI Package deployment

C.1 General

Annex C describes sequence examples of how to deploy different FDI Package types to different system architectures. The sequence and detailed features are system-specific.

For FDI Servers, no conditional deployment of UIP Variants is defined because package content can be consumed by different kinds of FDI Clients.

For standalone FDI host systems, FDI Server and FDI Client application are a single integrated entity. A standalone FDI host system can perform conditional deployment of the UIP Variants in accordance with the integrated FDI Client capabilities.

C.2 Scenarios

C.2.1 FDI Package deployment to PC based client/server systems

C.2.1.1 FDI Device Packages/FDI Profile Packages/FDI Communication Packages

The following steps apply for the deployment of FDI Device Packages, FDI Profile Packages and FDI Communication Packages to an FDI Server.

a) The user chooses an FDI Package from the file system.
b) The system validates the FDI Package signature and integrity.
c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies
   - that the PackageType shall be "Device", "Profile", or "Communication";
   - that the FDIVersionSupported shall be equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server;
   - the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by the FDI.
d) The system adds the FDI Package information to the system specific device catalog.
e) The system reads all UIPs stored in the FDI Package and adds all UIPs and all available variants to the system specific UIP catalog.
f) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.
g) The system stores the entire FDI Device Package content.
h) The system reads the EDD for each device type and creates Information Model (IM) type nodes.

C.2.1.2 FDI UIP Packages

The following steps apply for the deployment of UIP Packages to an FDI Server.

a) The user chooses an FDI Package from the file system.
b) The system validates the FDI Package signature and integrity.
c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies
   - that the PackageType is "Uip";

- that the FDIVersionSupported is equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and
- the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by the FDI.

d) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.

e) The system stores the entire FDI Device Package content.

f) The system reads all UIPs stored in the FDI Package and adds all UIPs and all available variants to the system specific UIP catalog.

C.2.2 FDI Package deployment to an FDI standalone system

C.2.2.1 FDI Device Packages/FDI Profile Packages/FDI Communication Packages

The following steps apply for the deployment of FDI Device Packages, FDI Profile Packages and FDI Communication Packages to an FDI standalone system.

a) The user chooses an FDI Package from the file system.

b) The system validates the FDI Package signature and integrity.

c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies

- that the PackageType is "Device", "Profile", or "Communication";
- that the FDIVersionSupported is equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and
- the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by FDI.

d) The system reads all UIP Variants for all UIPs in the FDI Package and verifies

- that the PlatformId and RuntimeId is supported by the integrated FDI Client; and
- the version and UipId against already imported UIP Variants and handles the update and upgrade accordingly. If there is already a higher version installed the import is aborted since downgrades are not supported by the FDI.

e) The system adds the UIP information of matching and imported UIPs to the system specific UIP catalog.

f) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.

g) The system stores the relevant FDI Device Package content.

h) The system reads the EDD for each device type and creates Information Model (IM) type nodes.

C.2.2.2 FDI UIP Packages

The following steps apply for the deployment of UIP Packages to an FDI standalone system.

a) The user chooses an FDI Package from file system.

b) The system validates the FDI Package signature and integrity.

c) The system reads the FDI Package Catalog information, as defined in 4.2.1:

- the PackageType attribute is being checked (see Annex E) for "Uip"
- the FDIVersionSupported shall be equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and

- the system imports UIPs including UIP Variants and checks for already imported versions of the particular UIP and handles the update and upgrade accordingly. If there is already a higher version installed, the import is aborted since downgrades are not supported by the FDI.

d) The system reads all UIP Variants for all UIPs in the FDI Package and verifies:
   - that the PlatformId and RuntimeId is supported by the integrated FDI Client; and
   - the version and UipId against already imported UIP Variants and handles the update and upgrade accordingly. If there is already a higher version installed, the import is aborted since downgrades are not supported by FDI.

e) The system stores the relevant FDI Device Package content.

Annex D
(informative)

Example

D.1 General

The purpose of Annex D is to provide an overview of ISO/IEC 29500-2 and an example FDI Device Package implementation. It is not intended to provide all the details that might be necessary for the implementation of an FDI Package.

Unless otherwise stated in Annex D, the ~~acronym~~ initialism OPC refers to Open Packaging Conventions.

D.2 Open Packaging Conventions

D.2.1 Overview

The Open Packaging Conventions (OPC) are a container-file technology specified in ISO/IEC 29500-2. OPC-based documents are ZIP archives that contain XML, binary and other types of files. They combine the advantages of maintaining the independence and the integrity of the files embedded in the document while having a single integrated package.

An OPC package consists of parts and relationships as shown in Figure D.1. Parts refer to the content being packaged such as binary and text files. Relationships define associations between the package, parts and external resources.

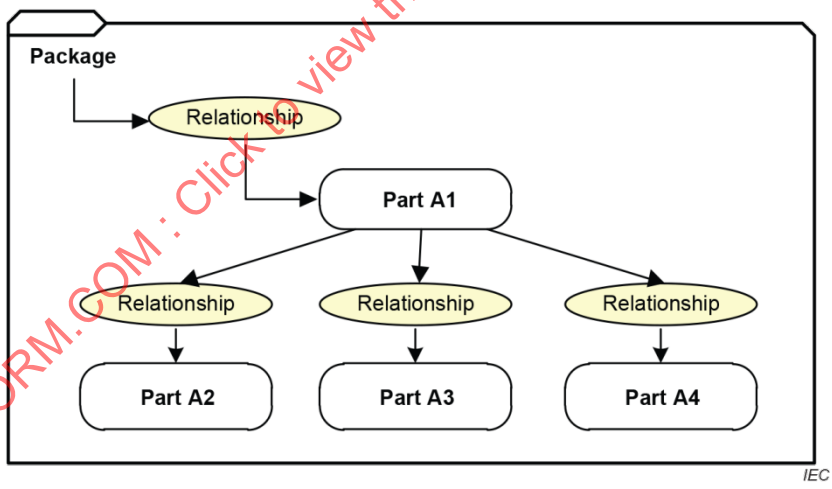

Figure D.1 – Parts and relationships in a package

D.2.2 Parts

OPC packages can store parts that contain any type of data (text, images, XML, binary, whatever). Parts can be organized as a hierarchy of folders that is similar to a file system. The OPC specification allows any folder organization that is convenient for the application.

Every part in a package has a unique URI-compliant part name along with a specified content-type expressed in the form of a MIME media type. The MIME media types for all the parts stored in the package are defined in an XML file named "/[Content_Types].xml".

In an FDI Package, the parts include the Package Catalog, EDD, UIPs and Attachments.

D.2.3 Relationships

In addition to a hierarchy of folders and parts, OPC allows the definition of relationships among parts. Relationships provide a layer of indirection, so dependencies are not hardcoded into the content.

Relationships allow problem domain and application specific relationships to be defined and enforced by systems and tools. By navigating through the network of associations, one can have random access to related content.

Relationships are composed of four elements:

- an identifier (ID);
- an optional source (the package or a part within the package);
- a relationship type (a URI-style expression that defines the type of the relationship);
- a target (a URI to another part within the package or to an external resource).

The relationships are stored in XML files with the extension ".rels" within subfolders named "/_rels".

In an FDI Package, the relationships are used to express the association of the package to the Package Catalog, the EDD, the UIP and the Attachments.

The following clause, for instance, defines the relationship identified as "rIdUip1", which establishes the association between the Package and the Package Catalog.

```
  <Relationship Id="rIdUip1"
    Type="http://fdi-cooperation.com/2010/relationships/package-
catalog"
    Target="fdipackage/catalog.xml"/>
```

D.2.4 OPC Core Features

D.2.4.1 Overview

The content types file /[Content_Types].xml, the package relationships file /_rels/.rels, and the part relationships files in the subfolders _rels are the core parts of an OPC package.

The filename [Content_Types].xml in any folder, the subfolder name _rels, and the file extension .rels within such directory are the only three reserved names for files stored in an OPC package.

D.2.4.2 Content Types

The file /[Content_Types].xml defines the MIME media types for all the parts stored in the package. It defines default mappings based on file extensions, along with overrides for specific parts with content types that are different from the file extension defaults.

Table D.1 and Table D.2 show examples of standard MIME media types and examples of FDI custom MIME media types that ~~may~~ can be used in FDI packages.

Table D.1 – Examples of standard MIME media types that can be used in FDI packages

Type of content	Standard MIME media type
XML file	application/xml
PNG image	image/png
PDF document	application/pdf
OPC relationships	application/vnd.openxmlformats-package.relationships+xml

Table D.2 – Examples of FDI custom MIME media types that can be used in FDI Packages

Type of content	FDI custom MIME media type
Package catalog	application/vnd.fdi.package.catalog+xml
EDD	application/vnd.fdi.package.edd
UIP	application/vnd.fdi.package.uip

The following clause in the /[Content_Types].xml file defines the default MIME media type for any XML files in the package.

```
<Default Extension="xml" ContentType="application/xml"/>
```

The following clause in the same file will override the definition above for the package catalog part only:

```
<Override PartName="/fdipackage/catalog.xml"
    ContentType="application/vnd.fdi.package.catalog+xml"/>
```

D.2.4.3 Package Relationships

The root level /_rels folder stores the relationships for the package as a whole. The /_rels folder normally contains a file named .rels, an XML file where the starting package-level relationships are stored.

D.2.4.4 Part Relationships

Each part may have its own relationships. If the part has relationships, they will be stored in an XML file within the _rels folder that is a sibling of that part. That file takes the part name with a .rels appended to it.

Because the Package Catalog part has relationships to other package parts, there is a file named catalog.xml.rels inside the /_rels folder that defines those relationships.

D.2.5 OPC Additional features

D.2.5.1 Core Properties

Core properties consist of package metadata. They enable users to get and set well-known and common sets of property metadata within packages, such as categorization of the content, status of package (e.g. draft, reviewed, final), date of creation, identification of the creator, keywords, and language. The core properties are not used in FDI Packages (see 5.2).

D.2.5.2 Thumbnails

Thumbnails are images that are used as graphical representation of parts of a package or a package as a whole. The use of thumbnails in an FDI Package is optional (see 5.2).

D.2.5.3 Digital Signatures

Digital signatures can be used to enable consumers to validate the integrity of the contents. The use of digital signatures is mandatory in FDI Packages (see 5.2).

D.3 Creation and Handling of FDI Packages

As long as the conventions are followed, OPC files can be created, opened and modified just as any ordinary ZIP file by using standard ZIP file tools. However, there exists higher level support to handle them so that developers do not have to cope with all the peculiarities of OPC. OPC is natively supported in Microsoft .NET Framework 3.0. Open source libraries also exist for other languages. Ultimately, FDI-specific tools are expected to provide automated support for the creation and handling of FDI Packages.

Given the catalog information, the EDD, the UIPs and attachments for a specific device type, a hypothetical FDI packaging tool is capable of creating the FDI package for that device type, as depicted in Figure D.2.

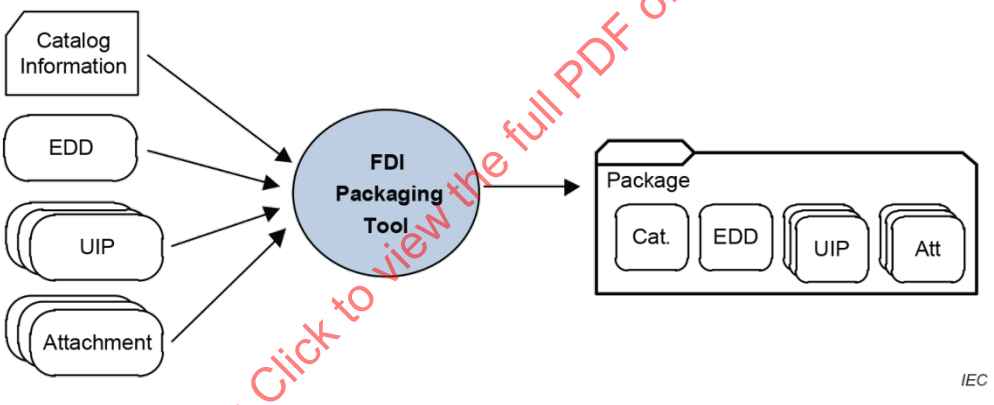

Figure D.2 – Creating an FDI Package with the content files

D.4 FDI Device Package Example

D.4.1 Overview

Figure D.3 represents an example FDI Device Package. The root directory of the FDI Device Package contains three directories and one file. The directory names fdipackage and uip are examples and are not defined by this document. The _rels directory is used to store the package relationships and is defined by ISO/IEC 29500-2.

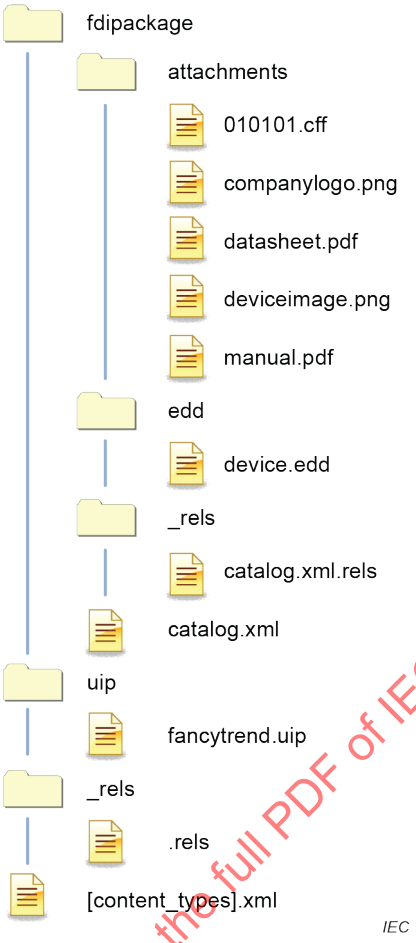

Figure D.3 – FDI Device Package Example

The file [Content_Types].xml is defined by ISO/IEC 29500-2 and defines the content type of the parts in the package. It is required for all packages. All parts in the package shall have a content type identified by this file.

An example for /[Content_Types].xml is listed below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
    <Default Extension="rels"
      ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
    <Default Extension="xml" ContentType="application/xml"/>
    <Default Extension="uip" ContentType="application/vnd.fdi.package.uip"/>
    <Default Extension="png" ContentType="image/png"/>
    <Default Extension="pdf" ContentType="application/pdf"/>
    <Default Extension="edd" ContentType="application/vnd.fdi.package.edd"/>
    <Default Extension="cff" ContentType="application/vnd.ff.cff"/>
    <Override PartName="/fdipackage/catalog.xml"
      ContentType="application/vnd.fdi.package.catalog+xml"/>
</Types>
```

In this example, the Package Catalog is identified by the part /fdipackage/catalog.xml with content type application/vnd.fdi.package.catalog+xml in the /[Content_Types].xml. The device has a protocol-specific CFF file. It is the responsibility of the protocol organization to define the content type associated with that file type. The content type for the CFF is an example and is not currently specified by the Fieldbus Foundation.

The /rels/.rels part defines the package relationships per ISO/IEC 29500-2.

An example for /rels/.rels is listed below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Relationships
  xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId1"
      Type="http://fdi-cooperation.com/2010/relationships/package-catalog"
      Target="fdipackage/catalog.xml"/>
    <Relationship
      Id="rIdUip1" Type="http://fdi-cooperation.com/2010/relationships/uip"
      Target="uip/fancytrend.uip"/>
</Relationships>
```

In accordance with this document, the Package Catalog shall be identified by a single package relationship. The FDI Server identifies the Package Catalog part by retrieving the part associated with this standard relationship type. In this example, the catalog relationship is represented by

```
<Relationship Id="rId1"
  Type="http://fdi-cooperation.com/2010/relationships/package-catalog"
  Target="fdipackage/catalog.xml"/>
```

The value for relationship id is not specified by this standard. ISO/IEC 29500-2 requires that each relationship have a unique relationship ID.

An FDI Device Package may also provide one or more UIPs. In accordance with this document, a UIP shall be identified by a package relationship. The FDI Server identifies a UIP in a package by searching for all parts with the associated relationship type. In this example, a single UIP is identified by

```
<Relationship Id="rIdUip1" Type="http://fdi-cooperation.com/2010/relationships/uip"
  Target="uip/fancytrend.uip"/>
```

An example for /fdicatalog/catalog.xml is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:Catalog
xmlns:fdi="http://fdi-cooperation.com/2010/package"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>ef377fd0-5de5-11df-a08a-0800200c9a66</PackageId>
  <PackageType>Device</PackageType>
  <Version>1.0.0</Version>
  <FdiVersionSupported>1.0.0</FdiVersionSupported>
  <ManufacturerName>ACME Transmitters</ManufacturerName>
  <ManufacturerContact>42 Wallaby Way, Sydney,
Australia</ManufacturerContact>
  <ManufacturerUrl>http://acme.local</ManufacturerUrl>
  <ManufacturerImage>rIdMfrLogo</ManufacturerImage>
  <ListOfDeviceTypes>
    <DeviceType>
      <Name>
        <value>Temperature Transmitter</value>
        <value xml:lang="fr">Transmetteur de température</value>
        <value xml:lang="de">Temperatur-Transmitter</value>
      </Name>
      <ClassificationId>SENSOR_TEMPERATURE</ClassificationId>
      <ListOfInterfaces>
        <Interface>
          <ListOfCommunicationProfiles>
            <CommunicationProfile>foundation_h1</CommunicationProfile>
          </ListOfCommunicationProfiles>
          <Version>5.0.0</Version>
          <Manufacturer>0000xff00</Manufacturer>
```

```
            <DeviceModel>0x1234</DeviceModel>
            <CommunicationRole>CLIENT</CommunicationRole>
            <ListOfCommunicationProfileSupportFiles>

<CommunicationProfileSupportFile>rIdCFF</CommunicationProfileSupportFi
le>
            </ListOfCommunicationProfileSupportFiles>
          </Interface>
        </ListOfInterfaces>
        <Edd>rIDEDD</Edd>
        <ListOfSupportedDeviceRevisions>
          <DeviceRevision>1.0.0</DeviceRevision>
        </ListOfSupportedDeviceRevisions>
        <ListOfImages>
          <Image>rIdPicture1</Image>
        </ListOfImages>
        <ListOfDocuments>
          <Document>rIdDocument1</Document>
          <Document>rIdDocument2</Document>
        </ListOfDocuments>
        <ListOfSupportedUips>
          <SupportedUip>
            <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
            <Name>Fancy Trend</Name>
            <Version>1.1.*</Version>
           <Optional>true</Optional>
          </SupportedUip>
        </ListOfSupportedUips>
      </DeviceType>
    </ListOfDeviceTypes>
</fdi:Catalog>
```

The Package Catalog part will reference other package parts through relationship IDs. Those relationship IDs will be defined in a relationship file name by the part and appended with the .rels suffix. In this example, this part is named fdipackage/_rels/catalog.xml.rels.

An example for /fdipackage/_rels/catalog.xml.rels is listed below. The relationship IDs are not defined by this document. In accordance with ISO/IEC 29500-2, these relationship IDs shall be unique. The ID names in this example were selected to better illustrate referencing.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/device.edd" Id="rIdEDD"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-image"
      Target="attachments/deviceimage.png" Id="rIdPicture1"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
      Target="attachments/datasheet.pdf" Id="rIdDocument1"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
      Target="attachments/manual.pdf" Id="rIdDocument2"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-protocol"
      Target="attachments/010101.cff" Id="rIdCFF"/>
</Relationships>
```

In this example, the product documentation is identified by the following.

```
<ListOfDocuments>
    <Document>rIdDocument1</Document>
    <Document>rIdDocument2</Document>
</ListOfDocuments>
```

The catalog identifies the files by the relationship ids that are found in the corresponding /fdipackage/_rels/catalog.xml.rels.

```
<Relationship
  Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
  Target="attachments/datasheet.pdf" Id="rIdDocument1"/>
<Relationship
  Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
  Target="attachments/manual.pdf" Id="rIdDocument2"/>
```

The example FDI Device Package identifies one supported UIP.

```
<SupportedUip>
    <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
    <Name>Fancy Trend</Name>
    <Version>1.1.*</Version>
</SupportedUip>
```

This document permits UIPs to be delivered in the FDI Device Package, or the UIPs may be delivered in a separate FDI UIP Package. In this example, the UIP is delivered in the FDI Device Package.

UIPs are also encoded using ISO/IEC 29500-2, but they are not considered a valid FDI Package type. An FDI server will not directly consume a UIP. A UIP shall be encoded in a valid FDI Package as shown in this example.

D.4.2 User Interface Plug-in

The following example is for the UIP referenced from the example FDI Package in Clause D.4. This UIP has two variants, one targeted for the workstation and the other for a mobile environment. The UIP part is encoded according to ISO/IEC 29500-2. Figure D.4 shows the structure of the example UIP. In this example, the filename for the UIP is fancytrend.uip, and is explicitly referenced in the package relationship ID in Clause D.4.

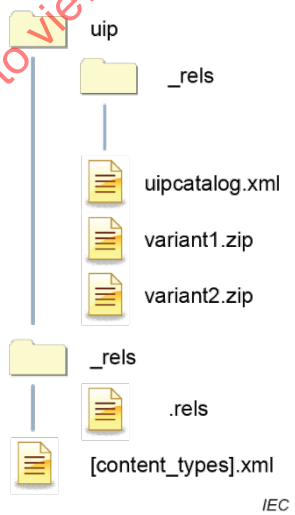

Figure D.4 – User Interface Plug-in Example (fancytrend.uip)

Similar to the example in Clause D.4, a package conforming to ISO/IEC 29500-2 shall have a /[content_types].xml and _rels/.rels part. In this example, the directory name uip is only an example and not specified by this document.

An example for [Content_Types].xml is listed below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
    <Default Extension="rels"
      ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
    <Default Extension="xml" ContentType="application/xml"/>
    <Default Extension="zip" ContentType="application/zip"/>
    <Override PartName="/uip/uipcatalog.xml"
      ContentType="application/vnd.fdi.package.uip.catalog+xml"/>
</Types>
```

An example for _rels/.rels part is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId1"
      Type="http://fdi-cooperation.com/2010/relationships/uip-catalog"
      Target="uip/uipcatalog.xml"/>
</Relationships>
```

In accordance with this document, the UIP Catalog part shall be identified by a package relationship. This is represented by the following code of the _rels/.rels part.

```
<Relationship Id="rId1"
  Type="http://fdi-cooperation.com/2010/relationships/uip-catalog"
  Target="uip/uipcatalog.xml"/>
```

The relationship ID is only an example. The relationship type allows an FDI Server to identify the part corresponding to the UIP Catalog. In this example, the catalog is the /uip/uipcatalog.xml part.

An example for /uip/uipcatalog.xml list listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:UipCatalog  xmlns:cat="http://fdi-cooperation.com/2010/package"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
    <Name>Fancy Trend</Name>
    <Version>1.0.0</Version>
    <FdiVersionSupported>1.0.0</FdiVersionSupported>
    <Additional>Best trend ever</Additional>
    <ListOfUIPVariants>
        <UIPVariant>
            <Variant>rIDVariant1</Variant>
            <Version>1.0.0</Version>
            <PlatformId>Workstation</PlatformId>
            <RuntimeId>String</RuntimeId>
            <StartElementName>Variant1.assembly</StartElementName>
        </UIPVariant>
        <UIPVariant>
            <Variant>rIDVariant2</Variant>
            <Version>1.0.0</Version>
            <PlatformId>Mobile</PlatformId>
            <RuntimeId>String</RuntimeId>
            <StartElementName>Variant2.assembly</StartElementName>
        </UIPVariant>
    </ListOfUIPVariants>
</fdi:UipCatalog>
```

~~The RunTimeId is defined in IEC 62769-6. As of this draft, those enumerations are not defined. The String should be replaced by the appropriate enumeration.~~

The UIP Catalog will have a corresponding relationship part. In this example, the part is /uip/_rels/uipcatalog.xml.rels.

An example for /uip/_rels/uipcatalog.xml.rels is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/uip-variant"
      Target="variant1.zip" Id="rIdVariant1"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/uip-variant"
      Target="variant2.zip" Id="rIdVariant2"/>
</Relationships>
```

In this example, the UIP Variants are stored in the /uip directory. It is possible that these could be stored in a different location. The target of the relationship would be updated to properly identify the UIP Variant.

D.4.3 EDD reference to UIP

The following EDD fragment can be used to identify the UIP in this example. The following code includes only the necessary attributes to establish the relationship. Other mandatory attributes have been removed for clarity.

```
MENU mymenu
{
    ITEMS
    {
        pFrancyTrend
    }
}

PLUGIN pFrancyTrend
{
    UUID f67e4ad0-5de5-11df-a08a-0800200c9a66;
}
```

D.4.4 FDI Registration Certificate

The example shows an FDI Device Package that describes device model 1234. The first registration was issued by FDI Registration Corp., Singapore in December 2010. A second registration was issued by FDI Registration Corp., ~~Köln~~Cologne in March 2011.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:FdiRegistrationCert
  xmlns:fdi=http://fdi-cooperation.com/2010/package
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>ef377fd0-5de5-11df-a08a-0800200c9a66</PackageId>
  <Version>1.3.4</Version>
  <ListOfRegDeviceTypes>
    <RegDeviceType>
      <Manufacturer>00ff00</Manufacturer>
      <DeviceModel>1234</DeviceModel>
      <ListOfRegistrations>
        <Registration>
          <RegistrationDate>2010-12-31</RegistrationDate>
          <RegAuthorityIdent>FDI Registration Corp.,
Singapore</RegAuthorityIdent>
          <RegistrationData>
            <value>Test Tool Version 2.1, Conformance Test<value>
            <value xml:lang="de">Test Tool Version 2.1, Conformance
Test</value>
          </RegistrationData>
        </Registration>
        <Registration>
          <RegistrationDate>2011-03-07</RegistrationDate>
```

```
          <RegAuthorityIdent>FDI Registration Corp.,
Köln</RegAuthorityIdent>
          <RegistrationData>
            <value>Test Tool Version 3.2, Conformance Test<value>
            <value xml:lang="de">Test Tool Version 3.2, Conformance
Test</value>
          </RegistrationData>
        </Registration>
      </ListOfRegistrations>
    </RegDeviceType>
  </ListOfRegDeviceTypes>
</fdi:FdiRegistrationCert>
```

The FDI Registration Certificate format does not only allow the description of several device types, but also the history listing of the registrations of a single type.

Annex E
(normative)

Schema

E.1 Target Namespace

The target namespace defined for the catalog document is defined by:

```
<xs:schema
   xmlns:fdi="http://fdi-cooperation.com/2010/package"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://fdi-cooperation.com/2010/package"
   elementFormDefault="unqualified" version="0.14.0">
```

E.2 Catalog

The Catalog element is the mandatory root element for the Package Catalog of an FDI Package.

The XML schema for a Catalog element is:

```
<xs:element name="Catalog" type="fdi:PackageT"/>
```

E.3 ClassificationIdT

The ClassificationIdT simple type specifies the device type classification (e.g. for sorted representation of installed device types represented by FDI Device Packages).

The XML schema for a ClassificationIdT is an enumeration and matches the discrete values of the CLASSIFICATION attribute specified in IEC 61804-3.

E.4 CommunicationProfileT

The CommunicationProfileT simple type specifies a unique identifier for the communication profile family and protocol using the pattern family_protocol. Valid ~~enumeration~~ values are specified in communication profiles (IEC 62769-1xx-x).

The XML schema for a CommunicationProfileT ~~enumeration~~ string type is:

```
<xs:simpleType name="CommunicationProfileT">
   <xs:restriction base="xs:string"/>
   <xs:enumeration value="foundation_h1"/>
         <xs:enumeration value="foundation_hse"/>
         <xs:enumeration value="hart_fsk"/>
         <xs:enumeration value="hart_psk"/>
         <xs:enumeration value="hart_wirelesshart"/>
         <xs:enumeration value="hart_ip"/>
         <xs:enumeration value="hart_rs485"/>
         <xs:enumeration value="hart_ir"/>
         <xs:enumeration value="profibus_dp"/>
         <xs:enumeration value="profibus_pa"/>
      <xs:enumeration value="profinet_io"/>
   </xs:restriction>
</xs:simpleType>
```

~~NOTE This schema is subject to be changed as soon as new protocols will be supported in FDI. The current set enumeration entries represent the current release status.~~

E.5 CommunicationRoleT

The CommunicationRoleT simple type specifies the supported communication function to differentiate if the device is a communication server, a gateway, or simple device type.

The XML schema for a CommunicationRoleT enumeration type is:

```
<xs:simpleType name="CommunicationRoleT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="SERVER"/>
      <xs:enumeration value="CLIENT"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a CommunicationRoleT enumeration type are described in Table E.1.

Table E.1 – Enumerations of CommunicationRoleT

Enumeration	Description
SERVER	Provides communication services for a specified protocol
CLIENT	Uses communication services implemented for a specified protocol

E.6 CommunicationServerT

The CommunicationServerT complex type specifies information to identify an FDI Communication Server.

The XML schema for a CommunicationServerT type is:

```
<xs:complexType name="CommunicationServerT">
  <xs:sequence>
    <xs:element name="ProductUri" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

The elements of a CommunicationServerT type are described in Table E.2.

Table E.2 – Elements of CommunicationServerT

Element	Description
ProductUri	ProductURI of the FDI Communication Sever

E.7 DeviceTypeT

The DeviceTypeT complex type specifies a device type definition.

The XML schema for a DeviceTypeT type is:

```
<xs:complexType name="DeviceTypeT">
   <xs:sequence>
      <xs:element name="Name" type="fdi:ListOfLocalizedStringsT"/>
      <xs:element name="ClassificationId"
         type="fdi:ClassificationIdT"/>
      <xs:element name="ListOfInterfaces"
         type="fdi:ListOfInterfacesT"/>
      <xs:element name="Edd" type="fdi:RelationshipIdT"/>
      <xs:element name="ListOfSupportedDeviceRevisions"
         type="fdi:ListOfSupportedDeviceRevisionsT" minOccurs="0"/>
      <xs:element name="ListOfImages" type="fdi:ListOfDeviceImagesT"
         minOccurs="0"/>
      <xs:element name="ListOfDocuments" type="fdi:ListOfDocumentsT"
         minOccurs="0"/>
      <xs:element name="ListOfSupportedUips"
         type="fdi:ListOfSupportedUipsT" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a DeviceTypeT type are described in Table E.3.

Table E.3 – Elements of DeviceTypeT

Element	Description
Name	Name of the device type The name can be localized
ClassificationId	Classification of the device type
ListOfInterfaces	List of interfaces supported by the device type
Edd	Reference to the EDD of the device type within the FDI Package
ListOfSupportedDeviceRevisions	List of compatible revisions of the device type that is described in this FDI Package
ListOfImages	List of references to images for this device type
ListOfDocuments	List of references to documents for this device type
ListOfSupportedUips	List of supported UIPs that are referenced by the EDD of this device type

E.8 FdiRegistrationCert

The FdiRegistrationCert element is the mandatory root element of the registration certificate.

The XML schema for a FdiRegistrationCert element is:

```
<xs:element name="FdiRegistrationCert"
   type="fdi:FdiRegistrationCertT"/>
```

E.9 FdiRegistrationCertT

The FdiRegistrationCertT complex type specifies the details of the registration certificate.

The XML schema for a FdiRegistrationCertT type is:

```
<xs:complexType name="FdiRegistrationCertT">
   <xs:sequence>
      <xs:element name="PackageId" type="fdi:UuidT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="ListOfRegDeviceTypes"
         type="fdi:ListOfRegDeviceTypesT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a FdiRegistrationCertT type are described in Table E.4.

Table E.4 – Elements of FdiRegistrationCertT

Element	Description
PackageId	A unique identifier for the FDI Package
Version	Package version
ListOfRegDeviceTypes	List of one registered device type

~~**E.10 HexStringT**~~

~~The HexStringT simple type specifies an identifier in hexadecimal.~~

~~The XML schema for a HexStringT type is:~~

```
<xs:simpleType name="HexStringT">
   <xs:restriction base="xs:string">
      <xs:pattern value="(0x)([0-9]|[A-F])+)"/>
   </xs:restriction>
</xs:simpleType>
```

E.10 InterfaceT

The InterfaceT complex type specifies elements of the interface supported by the device type.

The XML schema for an InterfaceT type is:

```
<xs:complexType name="InterfaceT">
   <xs:sequence>
      <xs:element name="ListOfCommunicationProfiles"
         type="fdi:ListOfCommunicationProfilesT"/>
      <xs:element name="Version">
         <xs:simpleType>
            <xs:restriction base="fdi:VersionT"/>
         </xs:simpleType>
      </xs:element>
      <xs:element name="Manufacturer" type="fdi:HexStringTxs:string"
         minOccurs="0"/>
      <xs:element name="DeviceModel" type="fdi:HexStringTxs:string"
         minOccurs="0"/>
      <xs:element name="CommunicationRole"
         type="fdi:CommunicationRoleT"/>
      <xs:element name="ListOfCommunicationProfileSupportFiles"
         type="fdi:ListOfProtocolSupportFilesT" minOccurs="0"/>
```

```
    </xs:sequence>
</xs:complexType>
```

The elements of an InterfaceT type are described in Table E.5.

Table E.5 – Elements of InterfaceT

Element	Description
ListOfCommunicationProfiles	List of communication profiles supported by the interface
Version	Version of the communication profile
Manufacturer	Manufacturer identifier as specified in the communication profile (IEC 62769-1xx-x) Shall be omitted in case of CommunicationRole = SERVER and shall be provided in case of CommunicationRole = CLIENT
DeviceModel	Device type identifier as specified in the communication profile (IEC 62769-1xx-x) Shall be omitted in case of CommunicationRole = SERVER and shall be provided in case of CommunicationRole = CLIENT
CommunicationRole	Supported communication functions for a specified protocol. A Communication Server shall describe at least one Interface element that contains CommunicationRole SERVER. A Communication Server shall not describe an Interface with CommunicationRole CLIENT. A Gateway shall describe at least one Interface element that contains CommunicationRole SERVER. A Gateway shall describe one Interface element that contains CommunicationRole CLIENT. A Device shall describe one Interface element that contains CommunicationRole CLIENT. A Device shall not describe an Interface with CommunicationRole SERVER.
ListOfCommunicationProfileSupportFiles	Optional list of communication profile support files

E.11 ListOfCommunicationProfilesT

The ListOfCommunicationProfilesT complex type is a list of one or more CommunicationProfiles.

The XML schema for a ListOfCommunicationProfilesT type is:

```
<xs:complexType name="ListOfCommunicationProfilesT">
    <xs:sequence>
        <xs:element name="CommunicationProfile"
            type="fdi:CommunicationProfileT" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

The elements of a ListOfCommunicationProfilesT type are described in Table E.6.

Table E.6 – Elements of ListOfCommunicationProfilesT

Element	Description
CommunicationProfile	Unique identifier for the communication profile. Valid values are listed in the communication profiles (IEC 62769-1xx)

E.12 ListOfDeviceImagesT

The ListOfDeviceImagesT complex type is a list of one or more images.

The XML schema for a ListOfDeviceImagesT type is:

```
<xs:complexType name="ListOfDeviceImagesT">
   <xs:sequence >
      <xs:element name="Image" type="fdi:RelationshipIdT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDeviceImagesT type are described in Table E.7.

Table E.7 – Elements of ListOfDeviceImagesT

Element	Description
Image	Reference to an image in the FDI Package

E.13 ListOfDeviceTypesT

The ListOfDeviceTypesT complex type is a list of one device type.

The XML schema for a ListOfDeviceTypesT type is:

```
<xs:complexType name="ListOfDeviceTypesT">
   <xs:sequence >
      <xs:element name="DeviceType" type="fdi:DeviceTypeT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDeviceTypesT type are described in Table E.8.

Table E.8 – Elements of ListOfDeviceTypesT

Element	Description
DeviceType	Device type definition

E.14 ListOfDocumentsT

The ListOfDocumentsT complex type is a list of one or more documents.

The XML schema for a ListOfDocumentsT type is:

```
<xs:complexType name="ListOfDocumentsT">
   <xs:sequence>
      <xs:element name="Document" type="fdi:RelationshipIdT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDocumentsT type are described in Table E.9.

Table E.9 – Elements of ListOfDocumentsT

Element	Description
Document	Reference to a document in the FDI Package

E.15 ListOfInterfacesT

The ListOfInterfacesT complex type is a list of one or more interfaces supported by the device type.

The XML schema for a ListOfInterfacesT type is:

```
<xs:complexType name="ListOfInterfacesT">
   <xs:sequence>
      <xs:element name="Interface" type="fdi:InterfaceT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfInterfacesT type are described in Table E.10.

Table E.10 – Elements of ListOfInterfacesT

Element	Description
Interface	Interface definition

E.16 ListOfLocalizedStringsT

The ListOfLocalizedStringsT complex type is a list of one or more strings localized by attributes.

The XML schema for a ListOfLocalizedStringsT type is:

```
<xs:complexType name="ListOfLocalizedStringsT">
   <xs:sequence>
      <xs:element name="value" type="fdi:LocalizedStringT"
         minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>

</xs:complexType>
```

The elements of a ListOfLocalizedStringsT type are described in Table E.11.

Table E.11 – Elements of ListOfLocalizedStringsT

Element	Description
value	Value for a localized string

E.17 ListOfProtocolSupportFilesT

The ListOfProtocolSupportFilesT complex type is a list of one or more protocol support files.

The XML schema for a ListOfProtocolSupportFilesT type is:

```
<xs:complexType name="ListOfProtocolSupportFilesT">
   <xs:sequence>
      <xs:element name="CommunicationProfileSupportFile"
         type="fdi:RelationshipIdT" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfProtocolSupportFilesT type are described in Table E.12.

Table E.12 – Elements of ListOfProtocolSupportFilesT

Element	Description
CommunicationProfileSupportFile	Reference to a communication profile support file in the package

E.18 ListOfRegDeviceTypesT

The ListOfRegDeviceTypesT complex type is a list of one registered device type.

The XML schema for a ListOfRegDeviceTypesT type is:

```
<xs:complexType name="ListOfRegDeviceTypesT">
   <xs:sequence>
      <xs:element name="RegDeviceType" type="fdi:RegDeviceTypeT"
         minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfRegDeviceTypesT type are described in Table E.13.

Table E.13 – Elements of ListOfRegDeviceTypesT

Element	Description
RegDeviceType	A registered device type

E.19 ListOfRegistrationsT

The ListOfRegistrationsT complex type is a list of one or more registered device types.

The XML schema for a ListOfRegistrationsT type is:

```
<xs:complexType name="ListOfRegistrationsT">
   <xs:sequence>
      <xs:element name="Registration" type="fdi:RegistrationT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfRegistrationsT type are described in Table E.14.

Table E.14 – Elements of ListOfRegistrationsT

Element	Description
Registration	Registration describing registration details as per the registration authority

E.20 ListOfSupportedDeviceRevisionsT

The ListOfSupportedDeviceRevisionsT complex type is a list of one or more device revisions that are compatible with this FDI Package.

The XML schema for a ListOfSupportedDeviceRevisionsT type is:

```
<xs:complexType name="ListOfSupportedDeviceRevisionsT">
   <xs:sequence>
      <xs:element name="DeviceRevision" type="fdi:VersionSupportedT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfSupportedDeviceRevisionsT type are described in Table E.15.

Table E.15 – Elements of ListOfSupportedDeviceRevisionsT

Element	Description
DeviceRevision	Revision of the device that is compatible with this device type in the FDI Package

E.21 ListOfSupportedUipsT

The ListOfSupportedUipsT complex type is a list of one or more User Interface Plug-ins.

The XML schema for a ListOfSupportedUipsT type is:

```
<xs:complexType name="ListOfSupportedUipsT">
   <xs:sequence>
      <xs:element name="SupportedUip" type="fdi:SupportedUipT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfSupportedUipsT type are described in Table E.16.

Table E.16 – Elements of ListOfSupportedUipsT

Element	Description
SupportedUip	Description of a UIP that is compatible with this device type

E.22 ListOfUipVariantsT

The ListOfUipVariantsT complex type is a list of one or more UIP variants.

The XML schema for a ListOfUipVariantsT type is:

```
<xs:complexType name="ListOfUipVariantsT">
   <xs:sequence>
      <xs:element name="UIPVariant" type="fdi:UipVariantT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfUipVariantsT type are described in Table E.17.

Table E.17 – Elements of ListOfUipVariantsT

Element	Description
UIPVariant	UIP Variant definition

E.23 LocalizedStringT

The LocalizedStringT complex type specifies localized string specified by the attribute. Strings with no language specification default to English.

The XML schema for a LocalizedStringT type is:

```
<xs:complexType name="LocalizedStringT">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="xml:lang" use="optional" default="en"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

The attributes of a LocalizedStringT type are described in Table E.18.

Table E.18 – Attributes of LocalizedStringT

Attribute	Description
lang	Language code according to ISO 639-1

E.24 PackageT

The PackageT complex type specifies the elements of the Package Catalog.

The XML schema for a PackageT type is:

```
<xs:complexType name="PackageT">
   <xs:sequence>
      <xs:element name="PackageId" type="fdi:UuidT"/>
      <xs:element name="PackageType" type="fdi:PackageTypeT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="FdiVersionSupported" type="fdi:VersionT"/>
      <xs:element name="ManufacturerName" type="fdi:String256T"/>
      <xs:element name="ManufacturerContact"
         type="fdi:LocalizedStringT" minOccurs="0"/>
      <xs:element name="ManufacturerUrl" type="xs:anyURI"
         minOccurs="0"/>
      <xs:element name="ManufacturerImage" type="fdi:RelationshipIdT"
         minOccurs="0"/>
     <xs:element name="CommunicationServer"
        type="fdi:CommunicationServerT" minOccurs="0"/>
      <xs:element name="ListOfDeviceTypes"
         type="fdi:ListOfDeviceTypesT" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a PackageT type are described in Table E.19.

Table E.19 – Elements of PackageT

Element	Description
PackageId	Unique identifier of the FDI Package
PackageType	Identifies the unique type of the FDI Package
Version	Unique version number for the FDI Package according to the FDI version convention
FdiVersionSupported	FDI Technology Version supported by the FDI Package
ManufacturerName	Manufacturer name of the FDI Package
ManufacturerContact	General contact information for the manufacturer of the FDI Package. Contact information can be localized
ManufacturerUrl	Website contact for the manufacturer of the FDI Package
ManufacturerImage	Reference to an image in the FDI package of the manufacturer's logo. The image shall be PNG format and have a resolution of 256 × 256
CommunicationServer	Information on how to discover an FDI Communication Server This information shall only be provided for packages of an FDI Communication Server.
ListOfDeviceTypes	List of device types available in the FDI Package. Mandatory for PackageTypes Device, Profile and Communication

E.25 PackageTypeT

The PackageTypeT simple type specifies the FDI Package type.

The XML schema for a PackageTypeT enumeration type is:

```
<xs:simpleType name="PackageTypeT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Device"/>
      <xs:enumeration value="Uip"/>
      <xs:enumeration value="Communication"/>
      <xs:enumeration value="Profile"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a PackageTypeT enumeration type are described in Table E.20.

Table E.20 – Enumerations of PackageTypeT

Enumeration	Description
Device	FDI Device Package
Uip	FDI UIP Package
Communication	FDI Communication Package
Profile	FDI Profile Package

E.26 PlatformT

The PlatformT simple type defines the target platform for the UIP Variant.

The XML schema for a PlatformT enumeration type is:

```
<xs:simpleType name="PlatformT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Workstation"/>
      <xs:enumeration value="Mobile"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a PlatformT enumeration type are described in Table E.21.

Table E.21 – Enumerations of PlatformT

Enumeration	Description
Workstation	Workstation platform
Mobile	Mobile platform

E.27 RegDeviceTypeT

The RegDeviceTypeT complex type specifies a device type definition.

The XML schema for a RegDeviceTypeT type is:

```
<xs:complexType name="RegDeviceTypeT">
   <xs:sequence>
      <xs:element name="Manufacturer" type="xs:string"/>
      <xs:element name="DeviceModel" type="xs:string"/>
      <xs:element name="ListOfRegistrations"
         type="fdi:ListOfRegistrationsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a RegDeviceTypeT type are described in Table E.22.

Table E.22 – Elements of RegDeviceTypeT

Element	Description
Manufacturer	Specific manufacturer identifier given by the communication profile rules
DeviceModel	Specific device type identifier given by the communication profile rules
ListOfRegistrations	Collection of Registrations with detailed registration information

E.28 RegistrationT

The RegistrationT complex type specifies details about the registration.

The XML schema for a RegistrationT type is:

```
<xs:complexType name="RegistrationT">
   <xs:sequence>
      <xs:element name="RegistrationDate" type="xs:date"/>
      <xs:element name="RegAuthorityIdent" type="xs:string"/>
      <xs:element name="RegistrationData"
         type="fdi:ListOfLocalizedStringsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a RegistrationT type are described in Table E.23.

Table E.23 – Elements of RegistrationT

Element	Description
RegistrationDate	Date when Registration Certificate was issued in format YYYY-MM-DD
RegAuthorityIdent	Identification information of the FDI Registration Authority (typically name and address)
RegistrationData	Data indicating the scope of the registration (description of test procedures, versions, validity, ...) – localized

E.29 RelationshipIdT

The RelationshipId simple type specifies the relationship ID in a part's relationship item for an embedded item within the FDI Package as specified in ISO/IEC 29500-2:2011, 9.3.2.

The XML schema for a RelationshipIdT type is:

```
<xs:simpleType name="RelationshipIdT">
   <xs:restriction base="xs:ID"/>
</xs:simpleType>
```

E.30 String256T

The String256T simple type specifies a string restricted to 256 or fewer characters.

The XML schema for a String256T type is:

```
<xs:simpleType name="String256T">
   <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
   </xs:restriction>
</xs:simpleType>
```

E.31 SupportedUipT

The SupportedUipT complex type specifies elements of a User Interface Plug-in.

The XML schema for a SupportedUipT type is:

```
<xs:complexType name="SupportedUipT">
   <xs:sequence>
      <xs:element name="UipId" type="fdi:UuidT"/>
      <xs:element name="Name" type="fdi:String256T"/>
      <xs:element name="Version" type="fdi:VersionSupportedT"/>
      <xs:element name="Optional" type="xs:boolean"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a SupportedUipT type are described in Table E.24.

Table E.24 – Elements of SupportedUipT

Element	Description
UipId	Identifier for a UIP
Name	Name of the UIP
Version	Versions of the UIP that are compatible with this FDI Package
Optional	If true, then the UIP is optional for the proper device integration. If false, then the UIP is required for proper device integration.

E.32 UipCatalog

The UipCatalog is the mandatory root element for the UIP Catalog of a User Interface Plug-in.

The XML schema for a UipCatalog element is:

```
<xs:element name="UipCatalog" type="fdi:UipT"/>
```

E.33 UipStyleT

The UipStyleT simple type specifies the style a UIP should run.

The XML schema for a UipStyleT enumeration type is:

```
<xs:simpleType name="UipStyleT" default="DIALOG">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WINDOW"/>
    <xs:enumeration value="DIALOG"/>
  </xs:restriction>
</xs:simpleType>
```

The enumeration values of a UipStyleT enumeration type are described in Table E.25.

Table E.25 – Enumerations of UipStyleT

Enumeration	Description
WINDOW	UIP should run as a modeless window If the parent starting the UIP is running modal, the UIP will run modal, otherwise modeless
DIALOG	UIP will always run as a modal window

E.34 UipT

The UipT complex type specifies the elements of the User Interface Plug-in catalog.

The XML schema for a UipT type is:

```
<xs:complexType name="UipT">
   <xs:sequence>
      <xs:element name="UipId" type="fdi:UuidT"/>
      <xs:element name="Name" type="fdi:String256T"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="FdiVersionSupported" type="fdi:VersionT"/>
      <xs:element name="Additional" type="fdi:String256T"/>
      <xs:element name="Style" type="fdi:UipStyleT "/>
      <xs:element name="ListOfUipVariants"
         type="fdi:ListOfUipVariantsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a UipT type are described in Table E.26.

Table E.26 – Elements of UipT

Element	Description
UipId	Unique identifier of the UIP
Name	Name of the UIP
Version	Version of the UIP
FdiVersionSupported	Version of the FDI Technology supported by this UIP
Additional	Additional information about the UIP
Style	Style of the UIP identifying where running modal or non-modal
ListOfUIPVariants	List of UIP Variants provided by this UIP

E.35 UipVariantT

The UipVariantT complex Type specifies the elements of a User Interface Plug-in variant.

The XML schema for a UipVariantT type is:

```
<xs:complexType name="UipVariantT">
   <xs:sequence>
      <xs:element name="Variant" type="fdi:RelationshipIdT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="PlatformId" type="fdi:PlatformT"/>
      <xs:element name="RuntimeId" type="xs:string"/>
      <xs:element name="CpuInformation" type="xs:string"
         minOccurs="0">
      <xs:element name="StartElementName" type="xs:string"/>
      <xs:element name="ListOfDocuments" type="fdi:ListOfDocumentsT"
         minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a UipVariantT type are described in Table E.27.

Table E.27 – Elements of UipVariantT

Element	Description
Variant	Reference to the UIP variant within the FDI Package
Version	Version of the UIP Variant
PlatformId	Platform of the UIP Variant
RuntimeId	Runtime environment of the UIP as specified in IEC 62769-6
CpuInformation	The element value provides additional information about the execution environment associated with the UIP runtime. The allowed values are specified in IEC 62769-6
StartElementName	Element that is loaded on an FDI Client to start the UIP Variant as specified in IEC 62769-6
ListOfDocuments	Optional list of references to documents for this UIP Variant

E.36 UuidT

The UuidT simple type specifies a universally unique identifier as specified by ISO/IEC 11578. The UuidT is restricted to the formal xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx where x is a hexadecimal digit.

The XML schema for a UuidT type is:

```
<xs:simpleType name="UuidT">
   <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
         [0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
   </xs:restriction>
</xs:simpleType>
```

E.37 VersionSupportedT

The VersionSupportedT simple type specifies version compatibility. Version can be explicitly referenced (e.g. 1.1.1) or the * wild card can be used to specify a range of compatibility (e.g. 1.1.*). Wildcards are permitted for minor release and revision.

The XML schema for a VersionSupportedT type is:

```
<xs:simpleType name="VersionSupportedT">
   <xs:restriction base="xs:string">
      <xs:maxLength value="16"/>
      <xs:pattern value="(\d+)\.((\d+\.(\d+|\*))|(\*\.\*))"/>
   </xs:restriction>
</xs:simpleType>
```

E.38 VersionT

The VersionT simple type specifies a version restricted to the format major.minor.revision.

The XML schema for a VersionT type is:

```
<xs:simpleType name="VersionT">
   <xs:restriction base="xs:string">
      <xs:maxLength value="16"/>
      <xs:pattern value="(\d+)\.(\d+)\.(\d+)"/>
   </xs:restriction>
</xs:simpleType>
```

Annex F
(normative)

Communication protocol specific profiles

Communication protocol specifics are described in so-called "Protocol-specific profile documents".

The creation and maintenance of those documents is the responsibility of the respective interest group representing the communication protocol (see Table F.1).

Table F.1 – Communication protocol interest groups (alphabetical order)

Protocol	Interest Group
FOUNDATION Fieldbus[a)]	~~Fieldbus FOUNDATION~~ FieldComm Group
HART[b)], WirelessHART[c)]	~~HART Communication Foundation~~ FieldComm Group
PROFIBUS[d)], PROFINET[e)]	PROFIBUS and PROFINET International
ISA100 [f)]	International Society of Automation (ISA)

a) In accordance with IEC 61784-1, CPF1.

b) In accordance with IEC 61784-1, CP9/1.

c) In accordance with IEC 62591.

d) In accordance with IEC 61784-1, CPF3.

e) In accordance with IEC 61784-2, CPF3.

f) In accordance with IEC 62734.

NOTE See the trade name declarations in IEC 61784-1 and IEC 61784-2.

Annex G
(informative)

FDI Package life cycle use cases

G.1 New Device Type

Adding new devices to a plant is a typical use case when a plant or part of a plant is being extended.

The new device type may be handled by an FDI Package that is already installed or it may need a new FDI Package that represents the new device type in the system. An already installed FDI Package might also be upgraded or a new specific FDI Package is required in order to support a new device type.

G.2 Replacement of Device

During the life cycle of a plant, it may happen that there is a need to repair or replace a device or device module. For example, in the case of Modular Devices, individual parts of the device are replaceable. Table G.1 shows device replacement guidelines.

Table G.1 – Device Replacement Guidelines

Device Manufacturer	Scenario	Guideline
Same manufacturer	Same type and same device version	—
Same manufacturer	Same type and new device version	May require an FDI Package update or upgrade
Same manufacturer	New type that is functionally compatible	May require an FDI Package update or upgrade or a new FDI Package
Same manufacturer	New type, that is functionally incompatible	Similar to that of the guideline for the new device type
Different manufacturer	Any	Similar to that of the guideline for the new device type

The device vendor shall communicate which FDI Package versions can be used for the new device revision. The vendor shall provide information on how to detect if the FDI Package shall be updated/upgraded and where to get this update/upgrade. Furthermore, the vendor shall provide a new or updated FDI Package version, if a device revision (Hardware) is incompatible with an already existing FDI Package version.

G.3 Firmware enhancements

An update of the device firmware may be required to fix errors that were detected in the firmware during the life cycle of the device. The device vendor provides maintenance of the device firmware by firmware updates, spare parts or replacement devices. Updating the device firmware is equivalent to the replacement of a device by the same type but new version (updated firmware).

The functionality of a device may be extended by upgrading the device firmware or changing the configuration of the device, for example, by installing a new device module. Table G.2 shows firmware enhancement guidelines.

Table G.2 – Firmware enhancement guidelines

Firmware	Guideline
Update	An update of firmware versions shall not require a new FDI Package version
Upgrade	In order to use this additional or enhanced functionality, an FDI Package upgrade may be required. Nevertheless the existing FDI Package version should be able to work with the new device firmware without leveraging the new device functionality

G.4 FDI Package life cycle polices

An FDI Package (including updates/upgrades) shall be shipped with product documentation including installation requirements, installation guideline, product description and release notes.

G.5 FDI Package update

A newer version of an FDI Package shall support all device types and data of the previous version of the FDI Package.

An FDI Package update should not change the scope of use. This includes:

- the available functionality (also user interface related);
- the supported device types;
- the supported bus protocols;
- the data items;
- the supported operating systems.

G.6 FDI Package upgrade

A newer version of an FDI Package shall support all device types and data of the previous version of the FDI Package.

An FDI Package upgrade might change the scope of use. This includes:

- the additional available functionality (also user interface related);
- the additional supported device types;
- the additional supported bus protocols;
- the additional data items;
- the additional supported operating systems.

G.7 FDI Package replacement/exchange

A system should implement an FDI Package replacement strategy in order to support replacement of an FDI Package by another FDI Package in a project. The system should provide a means to detect if an already installed FDI Package can handle the new device type or if a new FDI Package shall be installed.

G.8 FDI Package uninstallation

The FDI Package shall provide all means for an easy removal. An unsuccessful removal shall be indicated by the system.

Commonly used components shall be handled correctly (software parts and components shall be removed only, if no other references from other software programs exist). Existing FDI Package-specific files shall not be deleted automatically and shall be reusable in FDI Packages provided by the same vendor.

If an FDI Package, which is used and instantiated in a system, has been removed for any reason, the system shall be able to indicate which FDI Package is missing. It shall inform the user about removed FDI Packages and its supported device types: vendor, device name, type and version. The Device Instance data of the removed FDI Package shall not be deleted from the Information Model until the instances are removed by the user.

Annex H
(normative)

Health Status Method

H.1 Background

Many devices contain embedded intelligence to calculate diagnostic conditions. Other devices may have limited embedded processing and rely on external business logic processing to calculate device diagnostic conditions. Diagnostic data representation may be in various forms and may be influenced by the device communication profile.

H.2 Device Health Status model

The health status state provides a high level, consistent structured view to the current operating condition of a device independent of device or communication profiles. The health status state is calculated in an EDD method by accessing one or more device variables, calculating the health status state and returning a standard value to the application.

Some devices may offer configuration capability to map specific device diagnostic information to the health status state. The configuration of conditions to the health status state is device or communication profile-specific and is not part of this document.

The health status state shall be calculated in accordance with Table H.1. In the event of multiple conditions, the state with the lowest priority shall be returned.

Table H.1 – Health Status State

Health Status State	Priority	Conditions
Indeterminate	0	The health status is unavailable and therefore indeterminate. For example, the device may not be connected, a communication fault has occurred, or the device does not support the health status state.
Failure	1	Output signal is invalid due to malfunction in the field device or its peripherals.
Function Check	2	Output signal is temporarily invalid (e.g. frozen) due to ongoing work on the device.
Out of Specifications	3	Deviations from the permissible ambient or process conditions determined by the device itself through self-monitoring or faults in the device itself indicate that the measuring uncertainty of sensors or deviations from the set value in actuators is probably greater than expected under operating conditions.
Maintenance Required	4	Although the output signal is valid, the wear reserve is nearly exhausted, or a function will soon be restricted due to operational conditions.
Good	5	The device is operating under typical operating conditions such that Maintenance Requirement, Out of Specification, Failure and Function Check are not active.

H.3 Standard EDD Method signature

The EDD shall implement the GetHealthStatus method to provide access to the health status state. The method definition will be specific to the EDD. The method definition can use communication Builtins and shall not use user interface Builtins. See ~~IEC 61804-4:–, 7.1~~ IEC 61804-4:2020, 5.3 for a list of communication Builtins and user interface Builtins.

The GetHealthStatus method shall return the health status state priority value in accordance with Table H.1. Devices that do not support calculating the health status state shall return 0.

```
METHOD GetHealthStatus
{
   LABEL "GetHealthStatus";
   TYPE unsigned char;
   DEFINITION
   {
      /* device specific definition */

      /* return health status priority */
   }
}
```

For modular, block-oriented devices, multiple health statuses may be available. In this case, the method name shall use the prefix GetHealthStatus_ (e.g. METHOD GetHealthStatus_TB). Block-oriented health status methods shall be listed in the METHOD_ITEMS attribute of the associated BLOCK_A declaration.

H.4 Performance considerations

Accessing health status information via a standard EDD method requires business logic processing in the FDI Server. The method will typically require at least one communication access to the device to collect the health status. Continuous scanning of the health status across several device and device networks may have a serious impact on the performance of the underlying communication networks.

Underlying communication networks may provide optimized methods (e.g. asynchronous event driven messages) for obtaining health status information for continuous condition-based monitoring.

Annex I
(normative)

Modular devices

I.1 Concept

The concept of modular devices is shown in Figure I.1 and is as follows:

1) The entire modular device is described in a single package.
2) The device's modular structure and related configuration rules are described in a single EDD file. This EDD file represents the top-level topology element of the modular device's structure. This EDD file is referred in the catalog schema.
3) EDD files describing the modules are contained in separate EDD files, which are not exposed in the catalog XML. The reference to these modules' EDD files is made from the COMPONENT defined attribute named EDD.
4) Packaging of other package elements as it is defined in 4.2 is not touched.

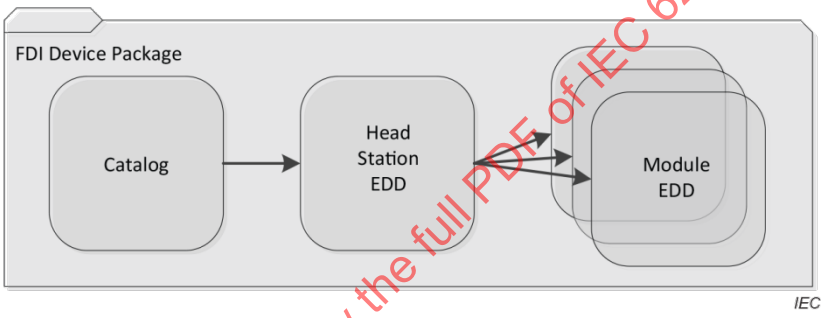

Figure I.1 – Modular device's package

I.2 EDDL usage profile

FDI Packages describing a modular device shall use the following EDDL defined constructs to describe the modular device's structure (topology) and related configuration rules:

1) COMPONENT
2) COMPONENT_FOLDER
3) COMPONENT_RELATION

The following EDDL defined syntax elements shall not be used:

1) COMPONENT_REFERENCE
2) INTERFACE
3) REQUIRED_INTERFACE
4) SUPPLIED_INTERFACE
5) FILTER

NOTE The rationale behind this decision is to reduce complexity for the FDI host implementation and for FDI Package creation. The restriction also protects the integrity of modular device description of one vendor since FDI does not support the extension of an existing modular device description with externally (other vendor) defined modules. This could happen if FDI supports using the EDDL defined syntax element COMPONENT_REFERENCE.

I.3 Processing recommendations

I.3.1 Monolithic device with device variants

This example shows how a pressure transmitter may be applied in different applications and for different measurement ranges.

The user places the top-level element in the topology. Now the host application can ask the user which device variant shall be used. (The same information can be read from the device based on the device vendor implemented "DETECT" function.) In order to define the actually needed device variant, the host application shall read the EDD and determine all COMPONENT and COMPONENT_FOLDER declarations.

I.3.2 Remote IOs

The user places the top-level element in the topology, which is the Remote IOs Head Station. Device variants have to be selected in accordance with the procedure described in ~~6.2.1~~ I.3.1.

For the purpose of the configuration of the module setup, the host needs to read the EDD and determines all COMPONENT, COMPONENT_FOLDER and COMPONENT_RELATION declarations to understand the device internal module catalog and the related configuration rules. The host can cache the device internal module catalog that is used only for the module configuration of this particular Head Station.

If Head Stations variants are described in separate EDDs, all these EDDs shall be referred to in the Catalog XML. These Head stations can share a common set of modules. The device internal module catalog shall be described in all Head stations EDDs. (This can be solved using "#include" in EDD source code).

I.3.3 How to identify the top-level topology element

All topology elements of the modular device are based on COMPONENT or COMPONENT_FOLDER declarations. The following text describes how an FDI host can find the topmost topology element inside an EDD file.

The FDI host has to find all COMPONENT declarations that do not use the EDD attribute. These COMPONENT declarations belong to internal hierarchy of the Head Station. The top-level declaration of this hierarchy can either be a COMPONENT or a COMPONENT_FOLDER. This top-level declaration corresponds to the device type described in the Catalog XML.

I.3.4 Packaging details example

Based on the description found in D.2.4 and D.4.1, I.3.4 provides additional information that helps to understand how the module EDD files need to be added beside the Head Station EDD file, which is also referred in the Catalog.XML file. The following example shows how three EDD files are integrated in a single package. There is one EDD file for the Head Station (Target="edd/HeadStation.edd") and two module EDD files (Target="edd/Module_A.edd" and Target="edd/Module_B.edd").

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/HeadStation.edd" Id="rIdEDD_HeadStation"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/Module_A.edd" Id="rIdEDD_Module_A"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/Module_B.edd" Id="rIdEDD_Module_B"/>
    …
</Relationships>
```

The entire set of EDD files can be found based on the specified relation type (Type = "http://fdi-cooperation.com/2010/relationships/edd").

The following catalog example is an excerpt to emphasize the concept of how the EDD file references work. The value calalog.xml defined element <EDD> refers to the package defined relation identifier (rIdEDD_HeadStation) that enables to retrieve the actual EDD file.

```
<DeviceType>
      <Name>
        <value>Modular remote IO</value>
       . . .
      </Name>
      <ClassificationId>REMOTEIO</ClassificationId>
      . . .
      <Edd>rIdEDD_HeadStation</Edd>
      . . .
</DeviceType>
```

Annex J
(normative)

FDI Communication Packages for FDI Communication Server

J.1 General

Details on packages for the different profiles are defined in Annex F. Annex J defines details on FDI Communication Packages used for the description and reference of FDI Communication Servers. They can be considered independent of technology profiles. This only considers the package, not the FDI Communication Server itself, which is defined in more detail in IEC 62769-7.

J.2 Protocol Support File

No additional file is required for FDI Communication Server packages.

J.3 CommunicationProfile definition

No values of CommunicationProfile are defined for FDI Communication Server packages.

J.4 Profile Device

There is no concept of a profile device for an FDI Communication Server.

J.5 Protocol version information

There is no product version information used for an FDI Communication Server.

J.6 Associating a Package with an FDI Communication Server

An OPC UA based FDI Communication Server is uniquely identified by its ProductUri. The mapping of the catalog information shall be in accordance with Table J.1.

Table J.1 – Catalog Mapping

Catalog Element	OPC UA Mapping
ProductUri	ProductUri

J.7 Handling of Catalog elements

Some parts of the catalog need to be handled in accordance with Table J.2.

Table J.2 – Handling of Catalog elements

Catalog Element	Handling
ClassificationId	"NETWORK"
ListOfSupportedDeviceRevisions	XML Element not provided

J.8 Example

An example for /fdicatalog/catalog.xml of an FDI Communication Server is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:Catalog
xmlns:fdi="http://fdi-cooperation.com/2010/package"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>f516f651-3e0f-4672-bcfe-67a4141a7a25</PackageId>
  <PackageType>Communication</PackageType>
  <Version>1.0.0</Version>
  <FdiVersionSupported>1.0.0</FdiVersionSupported>
  <ManufacturerName>Communication Provider GmbH</ManufacturerName>
  <ManufacturerContact>Hauptstrasse 17, Neustadt,
Germany</ManufacturerContact>
  <ManufacturerUrl>http://cpg.local</ManufacturerUrl>
  <ManufacturerImage>rIdMfrLogo</ManufacturerImage>
  <CommunicationServer>
    <ProductUri>urn:cpg:comserver</ProductUri>
  </CommunicationServer>
  <ListOfDeviceTypes>
    <DeviceType>
      <Name>
        <value>FDI Communication Server for HART</value>
        <value xml:lang="de">FDI Kommunikationsserver für HART</value>
      </Name>
      <ClassificationId>NETWORK</ClassificationId>
      <ListOfInterfaces>
        <Interface>
          <ListOfCommunicationProfiles>
            <CommunicationProfile>hart_fsk</CommunicationProfile>
          </ListOfCommunicationProfiles>
          <Version>5.0.0</Version>
          <CommunicationRole>SERVER</CommunicationRole>
        </Interface>
      </ListOfInterfaces>
      <Edd>rIDEDD</Edd>
      <ListOfImages>
        <Image>rIdPicture1</Image>
        <Image>rIdPicture2</Image>
      </ListOfImages>
      <ListOfDocuments>
        <Document>rIdDocument1</Document>
      </ListOfDocuments>
    </DeviceType>
  </ListOfDeviceTypes>
</fdi:Catalog>
```

Annex K
(normative)

FDI Profile for EDDs

K.1 Overview

Annex K describes rules that need to be applied to an EDD in order to fulfil the conformance to the FDI profile for EDDs. Annex K does not define new EDD concepts or constructs, but only defines that some optional constructs defined in the EDD specification are mandatory and some other concepts shall not be used in order to be compliant with the FDI profile for EDDs.

K.2 Entry Point to Online handling

The EDD shall contain at least one entry point for online handling (device_root_menu, diagnostic_root_menu, maintenance_root_menu or process_variables_root_menu).

K.3 Entry Point to Offline handling

The EDD shall contain at least one entry point for offline handling by providing the offline_root_menu.

K.4 Upload and Download

The EDD shall contain an upload menu (upload_from_device_root_menu or download_variables). The EDD shall contain a download menu (download_to_device_root_menu ~~or upload_variables~~). The upload and download menus shall not contain any user interactions (call to User Interface Builtins).

K.5 Initial Data Set

The EDD shall provide a valid initial data set for offline configuration without being connected to the device. There shall be at least one device variant where this configuration could be directly downloaded without modifications.

~~This can be achieved by using INITIAL_VALUEs in the EDD or by using the defaults of the data types.~~

This can be achieved by using mechanisms defined in EDDL (e.g. INITIAL_VALUE, DEFAULT_VALUE) or by using the defaults of the respective data types.

NOTE EDD offers additional concepts to create valid offline configurations, such as TEMPLATEs. Those can be used to create different variants of initial settings.

K.6 Method GetHealthStatus

The EDD shall include the GetHealthStatus method to provide access to health status state. See Annex H.

K.7 Actions

K.7.1 Pre- and Post-Read Actions

The pre- and post-read actions (PRE_READ_ACTIONS and POST_READ_ACTIONS) on VARIABLEs or MENUs shall not contain any user interactions (call to User Interface Builtins).

K.7.2 Pre- and post-write actions

The pre- and post-write actions (PRE_WRITE_ACTIONS and POST_WRITE_ACTIONS) on VARIABLEs or MENUs shall not contain any user interactions (call to User Interface Builtins).

K.7.3 Refresh actions on variables

The refresh actions (REFRESH_ACTIONS) on VARIABLEs shall not contain any user interactions (call to User Interface Builtins).

NOTE Other refresh actions (e.g. on graphs) ~~may~~ can have calls to User Interface Builtins.

K.7.4 Actions on BIT_ENUMERATION

Actions on BIT_ENUMERATION shall not contain any user interactions (call to User Interface Builtins).

K.8 Shared files

Use of shared files (using SHARED on the FILE construct) is not recommended and will be ignored in FDI Hosts.

NOTE Future versions of the FDI Technology may support this feature.

Bibliography

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

~~IEC 61804-5, *Function blocks (FB) for process control and Electronic device description language (EDDL) – Part 5: EDDL Builtin library*~~

IEC 62591, *Industrial communication networks – Wireless communication network and communication profiles – WirelessHART*

~~FDI-2021, *FDI Project Technical Specification – Part 1: Overview* <available at www.fdi-cooperation.com>~~

~~FDI-2022, *FDI Project Technical Specification – Part 2: FDI Client* <available at www.fdi-cooperation.com>~~

~~FDI-2023, *FDI Project Technical Specification – Part 3: FDI Server* <available at www.fdi-cooperation.com>~~

~~FDI-2024, *FDI Project Technical Specification – Part 4: FDI Packages* <available at www.fdi-cooperation.com>~~

~~FDI-2025, *FDI Project Technical Specification – Part 5: FDI Information Model* <available at www.fdi-cooperation.com>~~

~~FDI-2026, *FDI Project Technical Specification – Part 6: FDI Technology Mapping* <available at www.fdi-cooperation.com>~~

~~FDI-2027, *FDI Project Technical Specification – Part 7: FDI Communication Devices* <available at www.fdi-cooperation.com>~~

IEC 62769-2, *FDI Project Technical Specification – Part 2: FDI Client*

IEC 62769-3, *FDI Project Technical Specification – Part 3: FDI Server*

IEC 62769-5, *FDI Project Technical Specification – Part 5: FDI Information Model*

IEC 62769-7, *FDI Project Technical Specification – Part 7: FDI Communication Devices*

NE107, *Self-Monitoring and Diagnosis of Field Devices,* available at www.namur.de

IEC 62769-4

Edition 2.0 2021-02

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Field device integration (FDI) – Part 4: FDI Packages

Intégration des appareils de terrain (FDI) – Partie 4: Paquetages FDI

IEC 62769-4:2021-02(en-fr)

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

FIELD DEVICE INTEGRATION (FDI) –

Part 4: FDI Packages

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62769-4 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) support for Package Developers to build EDDs targeted for today's EDD bases system under a single development tool;
b) digital signature now includes trusted timestamping for long-term validation of FDI Package;
c) time stamp for device package signature.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
65E/761/FDIS	65E/771/RVD

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

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

A list of all parts in the IEC 62769 series, published under the general title *Field Device Integration (FDI)*, can be found on the IEC website.

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

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

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

INTRODUCTION

The IEC 62769 series has the general title *Field Device Integration (FDI)* and the following parts:

– Part 1: Overview
– Part 2: FDI Client
– Part 3: FDI Server
– Part 4: FDI Packages
– Part 5: FDI Information Model
– Part 6: FDI Technology Mapping
– Part 7: FDI Communication Devices
– Part 100: Profiles – Generic Protocol Extensions
– Part 101-1: Profiles – Foundation Fieldbus H1
– Part 101-2: Profiles – Foundation Fieldbus HSE
– Part 103-1: Profiles – PROFIBUS
– Part 103-4: Profiles – PROFINET
– Part 109-1: Profiles – HART and WirelessHART
– Part 115-2: Profiles – Protocol-specific Definitions for Modbus RTU
– Part 150-1: Profiles – ISA 100.11a

FIELD DEVICE INTEGRATION (FDI) –

Part 4: FDI Packages

1 Scope

This part of IEC 62769 specifies the FDI Packages. The overall FDI architecture is illustrated in Figure 1. The architectural components that are within the scope of this document have been highlighted in Figure 1.

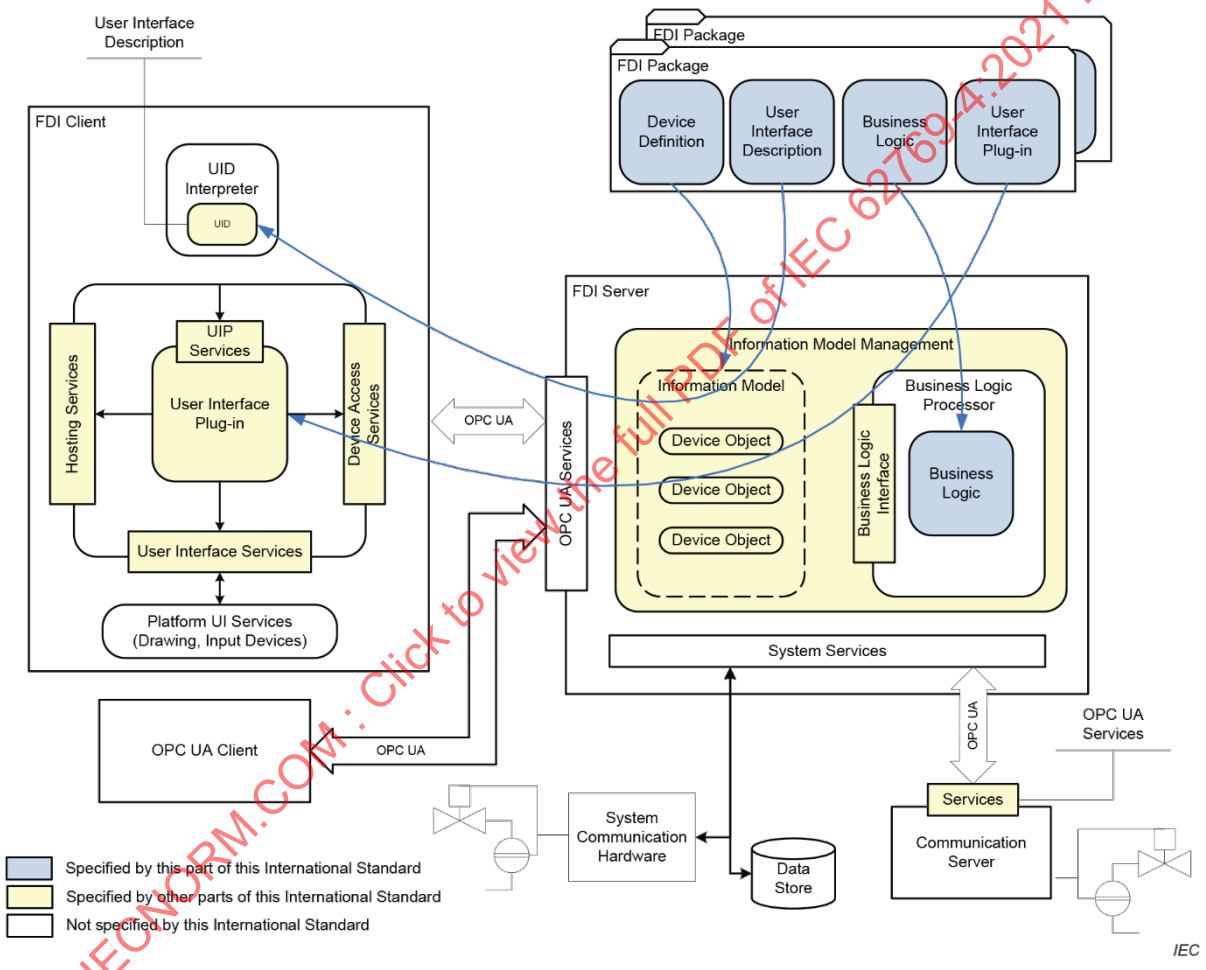

Figure 1 – FDI architecture diagram

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61804 (all parts), *Function blocks (FB) for process control and electronic device description language (EDDL)*

IEC 61804-5:2015, *Function blocks (FB) for process control and electronic device description language (EDDL) – Part 5: EDDL Builtin library*

IEC 62769-1, *Field Device Integration (FDI) – Part 1: Overview*

IEC 62769-6, *Field Device Integration (FDI) – Part 6: FDI Technology Mapping*

ISO/IEC 29500-2:2016, *Information technology – Document description and processing languages – Office Open XML File Formats – Part 2: Open Packaging Conventions*

ISO 639-1, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 32000-1, *Document management – Portable document format – Part 1: PDF 1.7*

Extensible Markup Language (XML) 1.0, W3C Recommendation, available at <http://www.w3.org/TR/REC-xml/>

XML Schema Definition Language (XSD) 1.1, W3C Recommendation, available at <http://www.w3.org/TR/xmlschema11-1/>

ETSI EN 319 132-1, *Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures*

ETSI TS 101 733, *Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)*

FIPS 140-2, *Security Requirements for Cryptographic Modules*

3 Terms, definitions, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1, ISO/IEC 29500-2, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
Attachment
device and protocol-specific support files that are not directly used to integrate the Device into the system

3.1.2
FDI Communication Package
FDI Package that provides information to integrate an FDI Communication Server to an FDI Server

3.1.3
FDI Device Package
FDI Package that provides one or more device types to an FDI Server

3.1.4
FDI Package Model
description of the structure and elements of an FDI Package

3.1.5
FDI Profile Package
FDI Package that provides information for creating a device type node that can be associated with a class of devices

3.1.6
FDI Registration Authority
entity that has the right and the ability to perform FDI conformance tests on FDI Packages and to issue registration certificate documents

3.1.7
FDI UIP Package
FDI Package that provides one or more UIPs to an FDI Server

3.1.8
Package Catalog
file that describes the contents of an FDI Package

3.1.9
UIP Catalog
file that describes the properties of a UIP

3.1.10
UIP Variant
platform-specific element of a User Interface Plug-in

Note 1 to entry: A UIP is composed of one or more variants. For example, one variant may be optimized for portable devices while another variant is optimized for large-screen devices.

3.2 Abbreviated terms

For the purposes of this document, the abbreviated terms given in IEC 62769-1 as well as the following apply.

CFF	Capabilities File for FOUNDATION Fieldbus
ID	Identifier
IDE	Integrated Development Environment
IM	Information Model
PNG	Portable network graphics
ZIP	Zipper (archive file format)

3.3 Conventions

For the purposes of this document, the conventions given in IEC 62769-1 apply.

4 FDI Package Model

4.1 Overview

The FDI Package Model (see Figure 2) provides all the elements necessary to integrate devices, network components and FDI Communication Servers into a system.

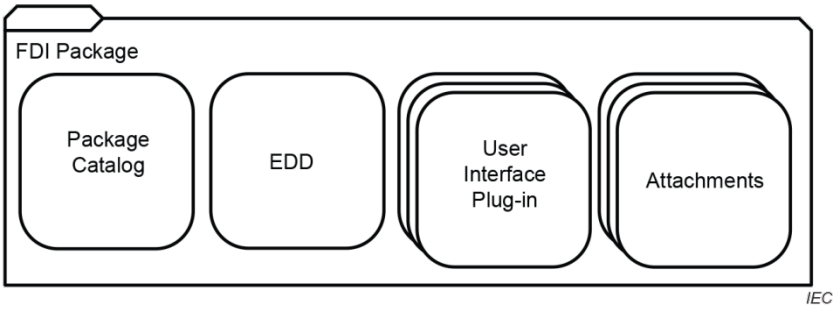

Figure 2 – FDI Package Model

Figure 3 shows the mapping of the FDI Package functional elements, as specified in IEC 62769-1, to the physical elements in an actual FDI Package, as specified in this document.

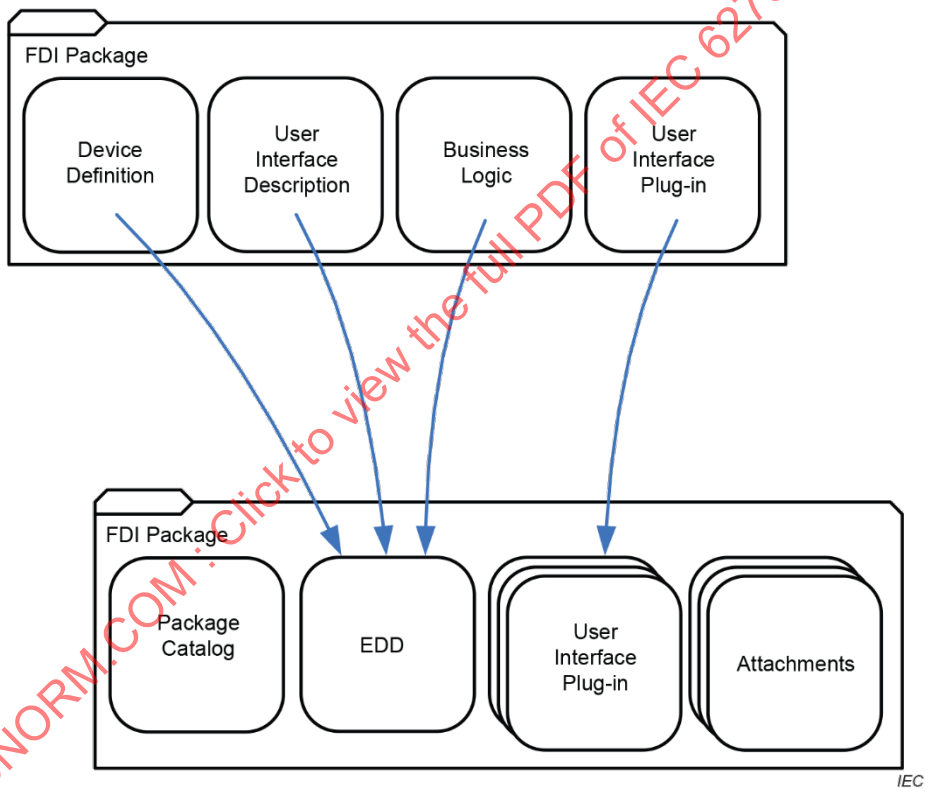

Figure 3 – Architectural mapping

The Electronic Device Description (EDD) corresponds to the Device Definition, the User Interface Description, and the Business Logic functional elements. A set of physical User Interface Plug-ins corresponds to the User Interface Plug-in functional element.

The other physical elements in the FDI Package, the Package Catalog and Attachments, provide support for basic mechanisms such as identification, versioning and deployment, and distribution of manufacturer and/or protocol-specific information about the device and/or the User Interface Plug-ins.

4.2 FDI Package Elements

4.2.1 Package Catalog

The Package Catalog is a required element that provides information about the contents of the FDI Package including, but not limited to, identification and version information, device type information, hardware and input/output device requirements, FDI Technology Version, and protocol-specific characteristics.

The Package Catalog is used by an FDI Server to create a catalog of device types and to create DeviceType Nodes in the Information Model.

4.2.2 Electronic Device Description

The EDD is an element that provides Device Definition, User Interface Descriptions, and Business Logic to an FDI Server.

Subclause 4.3 specifies for which FDI Package types an EDD is required.

To maximize interoperability, the initial setup of a device shall be achievable solely with the User Interface Descriptions, Device Definition and Business Logic that are part of the EDD in the FDI Device Package. The use of User Interface Plug-ins is optional and targeted in particular for the complete setup of complex devices.

4.2.3 User Interface Plug-in

A User Interface Plug-in (UIP) is an element that enables an FDI Client to present a programmed user interface. The FDI Server only stores the UIP from a consumed package; it does not execute or interpret the UIP.

A UIP is referenced from a User Interface Description. Those references are reflected in the Information Model through functional groups (see IEC 62769-5). References to UIPs in the Information Model are logical through a unique reference. The physical structure of the UIP is not at the Information Model level. Physically, a single UIP may consist of one or more UIP Variants, each targeted at a specific platform and run-time environment.

Figure 4 illustrates the logical reference to a UIP in the EDD and the physical structure of that UIP. The EDD of the FDI Package references a globally unique identifier of the UIP. In addition, the Package Catalog also lists the UIPs required by the device type along with the versions of the UIP supported by the package.

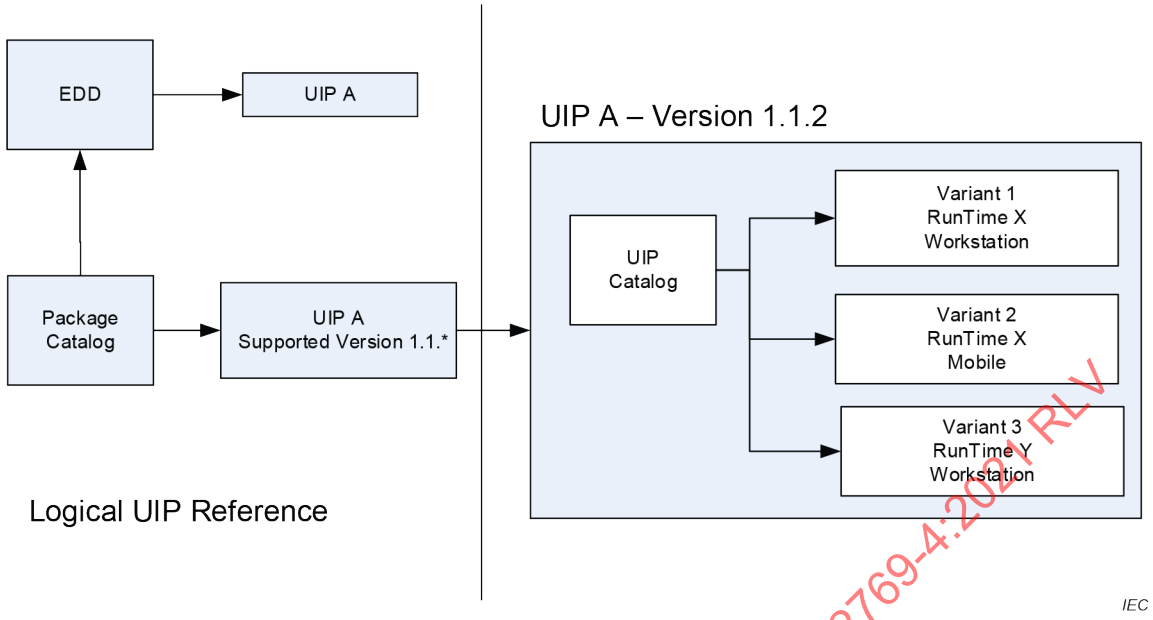

Figure 4 – User Interface Plug-in Reference Model

A UIP can be designed to meet different platform requirements since not all platforms support the same screen sizes and input devices.

When an FDI Client requests a UIP specifying the platform type, the deployment mechanism in the FDI Server looks for a UIP Variant that meets the platform type specified and returns it to the FDI Client.

The target platform defines specific screen resolutions and input devices that shall be supported by the UIP Variant. Available target platforms are described in Table 1.

Table 1 – UIP Platform

Platform	Characteristics
Workstation	A full client typically with a larger display, full keyboard and mouse input.
Mobile	A limited client typically with a smaller display and limited input capabilities.

Detailed platform requirements are technology dependent and specified in IEC 62769-6.

A UIP can be delivered internally (self-contained) or the UIP can be delivered through an FDI UIP Package. An EDD of one package shall not reference any UIP delivered self-contained with another package.

Figure 5 illustrates two FDI Device Packages referencing UIPs from an FDI UIP Package. In this example, the EDD from Device Package A and Device Package B references UIP B delivered by UIP Package C.

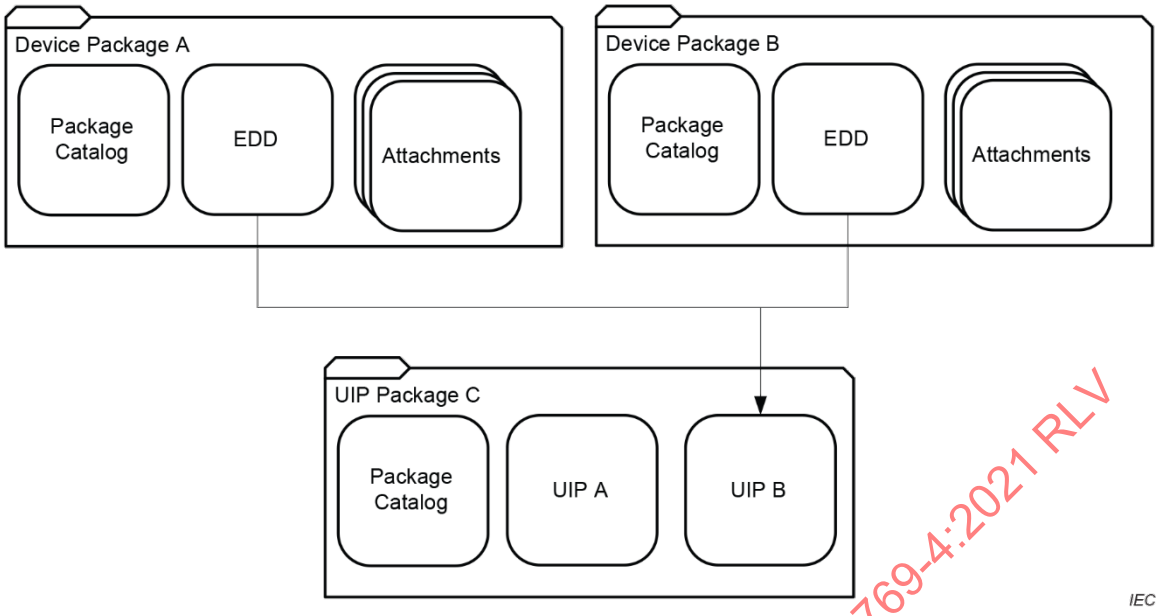

Figure 5 – Multiple FDI Packages referencing a common UIP

4.2.4 Attachment

Attachments provide device and protocol-specific support files, along with other files that are not directly used to integrate the device into the system, for example, product manuals.

The following types of Attachments are defined and reflected in the Package Catalog:

- Protocol-specific files (see 5.3.4.3);
- Documentation (see 5.3.4.2);
- Device pictures (see 5.3.4.10).

Some protocol-specific files are mandatory (see Annex F).

4.3 FDI Package Types

4.3.1 FDI Device Package

The FDI Device Package is intended to provide information about a device to a system. An FDI Device Package describes a single device type. Figure 6 shows the physical structure of an FDI Device Package describing a single device type. Details about how to create a package for a modular device is described in Annex I.

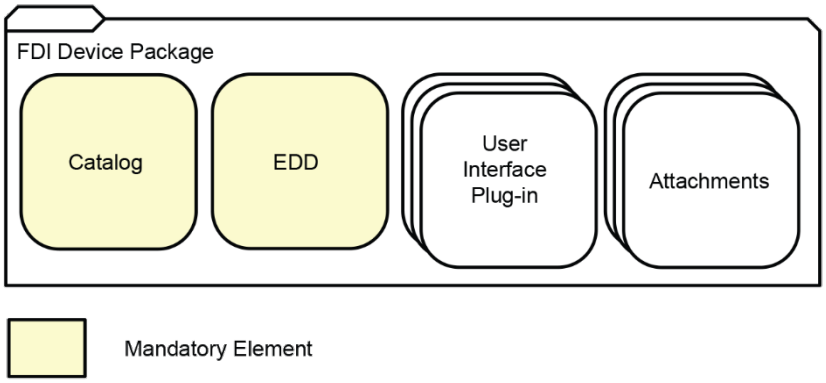

Figure 6 – FDI Device Package

The FDI Device Package shall have a single Package Catalog.

An FDI Package for simple Devices contains one EDD. For modular Devices, multiple EDDs may exist (see Annex I).

The FDI Device Package may include one or more User Interface Plug-ins.

The FDI Device Package may include Attachments.

4.3.2 FDI Communication Package

The FDI Communication Package is intended to provide information about a single FDI Communication Device. The group of FDI Communication Devices splits into two sub-groups.

– FDI Communication Packages for Gateways – Contain everything required to describe all device functions and logic required to bridge between different communication protocol networks, including the bridging algorithms (details are described in IEC 62769-7).

– FDI Communication Packages for FDI Communication Servers – Contain basic information to describe the communication device and to integrate it into the FDI Host but also to create a valid reference to the external FDI Communication Server application. This application shall not be delivered as part of the FDI Communication Package. The FDI Communication Server provides access to single field devices or field device networks (details are described in IEC 62769-7).

  Requirements to the FDI Communication Package content that are specific for FDI Communication Servers are defined in Annex J.

However, representatives of both groups are integrated into FDI Hosts by using FDI Communication Packages. Their physical structure is shown in Figure 7. Binary components necessary to communicate to the communication hardware shall be provided outside the scope of this FDI Communication Package.

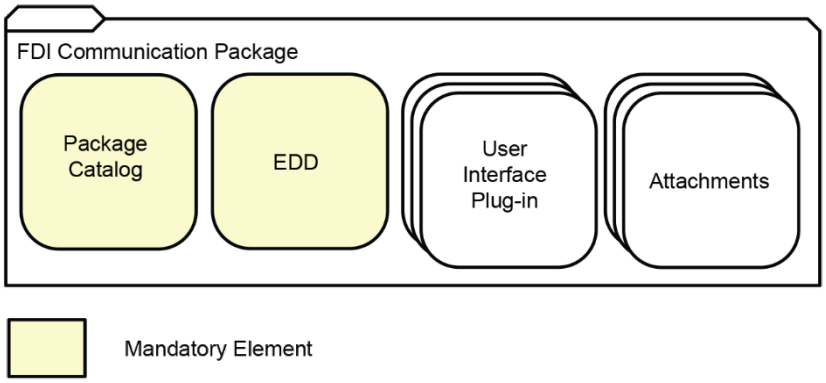

Figure 7 – FDI Communication Package

4.3.3 FDI UIP Package

The FDI UIP Package delivers User Interface Plug-ins to an FDI Server. It is used to distribute one or more UIPs that are intended to be shared by several device types. Its physical structure is illustrated in Figure 8.

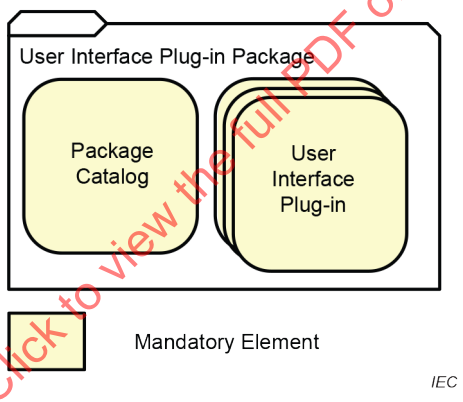

Figure 8 – FDI UIP Package

The FDI UIP Package shall have a Package Catalog and one or more User Interface Plug-ins.

4.3.4 FDI Profile Package

An FDI Profile Package provides information for creating a device type node that can be associated with a class of devices implementing a common set of parameters and functions (described in profiles or profile like definitions created by, for example, communication foundations or similar interest groups).

Conceptually, an FDI Profile Package provides information that is a super class of an FDI Device Package. The FDI Profile Package is represented in Figure 9.

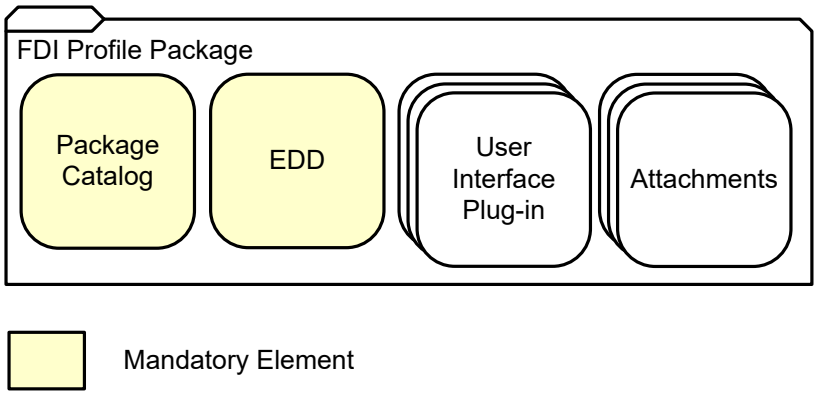

Figure 9 – FDI Profile Package

These packages enable integration of devices without having a specific FDI Device Package. FDI Profile Packages only support standard functionalities as defined by the corresponding communication protocol and application profiles (Annex F).

An FDI Profile Package shall be self-contained. For example, if a Profile B extends the definition of a Profile A, then the FDI Profile Package for the Profile B shall include a complete definition of Profile B, including the elements from Profile A.

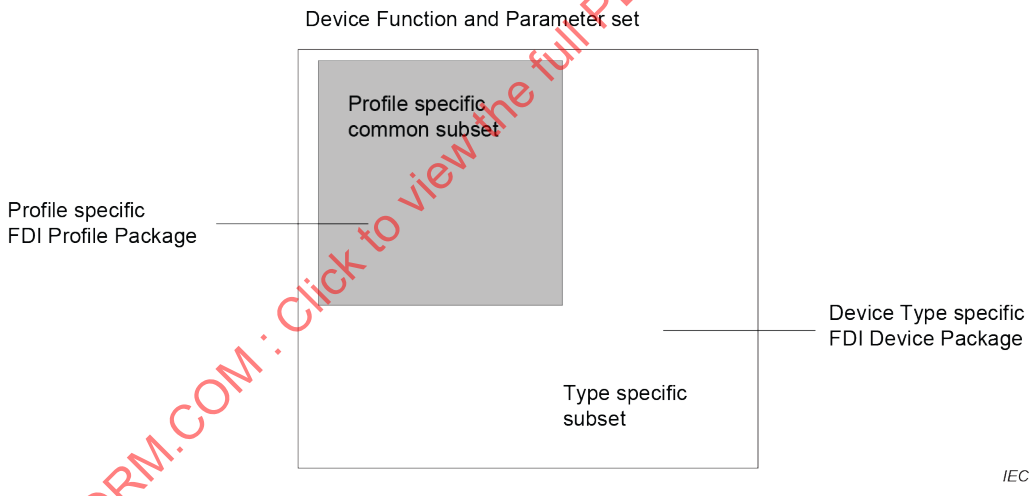

Figure 10 – Device Function and Parameter sets (type- and profile-specific)

Figure 10 represents the entire functionality of a device. Some of the functionality conforms to a specific communication profile, which may be accessed by using an FDI Profile Package. If there is a need to access Device Type specific functionality, in addition to the common functionality, then an FDI Device Package is required.

The detailed description of the communication protocol-specific requirements is not part of this document.

An automatic FDI Package selection implementation (launching FDI Packages matching a connected physical device) typically would search first for available Device Type specific FDI Device Packages. If the specific package is not available, the profile specific FDI Profile Package is loaded (if available). This behavior, however, is system-specific.

5 FDI Package implementation

5.1 Packaging technology

The FDI Package's format complies with the Open Packaging Convention as specified by ISO/IEC 29500-2. The Open Packaging Convention specification is designed to represent a broad range of applications. The technology was designed to provide a mechanism for delivering, deploying, and utilizing a set of logically and physically related components in a flexible, extensible, efficient, and open manner.

Subclause 5.2 clarifies the use of specific elements of the Open Packaging Conventions specification.

The naming conventions for the FDI Package file are described in Annex A. FDI Package creation fundamentals are described in Annex B. An example FDI Device Package implementation is described in Annex D.

NOTE *fdi-cooperation.com* was used when the FDI Cooperation LLC existed. The URL has not been changed to not endanger consistency. The URL does not indicate the current technology ownership and may not point to a valid address.

5.2 Use of Open Packaging Conventions

5.2.1 Unknown parts

Unknown parts are all parts not defined in this document or ISO/IEC 29500-2. Unknown parts may exist as the FDI specification is updated and the legacy FDI Server may not have knowledge of these additional parts. Unknown parts shall be ignored by the FDI Server to enable forward compatibility.

5.2.2 Invalid parts

Invalid parts are those parts that do not conform to the naming guidelines or that do not associate with content types specified in this document or ISO/IEC 29500-2. Invalid parts shall not be used in an FDI Package.

5.2.3 Unknown relationships

Unknown relationships are those relationships that are not defined in this document or ISO/IEC 29500-2. Unknown relationships shall be ignored by an FDI Package consumer.

5.2.4 Interleaving

All parts of the FDI Package shall be laid out using simple ordering. The package producer shall not interleave parts.

5.2.5 Core properties

The core properties are not used in the FDI Package Model and shall be ignored by all FDI Package consumers.

5.2.6 Thumbnails

The thumbnail is an optional part of an FDI Package.

5.2.7 Digital signatures

The use of digital signatures as specified in ISO/IEC 29500-2 is a mandatory part of an FDI Package. For more details, see Clause 7.

5.3 FDI Package Parts

5.3.1 Package Catalog

5.3.1.1 Format

The Package Catalog part is an XML file whose schema is defined in Annex E. An FDI Package shall have only one Package Catalog. The Package Catalog shall be identified by a single package relationship. The root element of the file is Catalog. The structure of a Catalog element is illustrated in Figure 11.

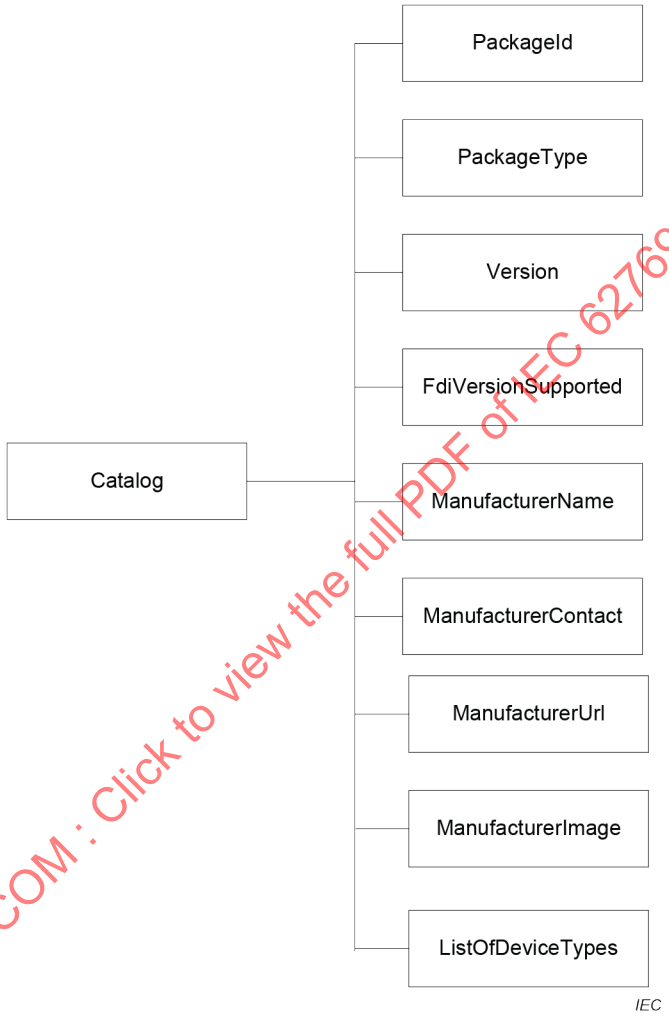

Figure 11 – Catalog Element

The Package Catalog part is described in Table 2.

Table 2 – Package Catalog Part

Part	Content
Content Type	application/vnd.fdi.package.catalog+xml
Root Namespace	http://fdi-cooperation.com/2010/package-catalog
Source Relationship	http://fdi-cooperation.com/2010/relationships/package-catalog
Filename	catalog.xml

5.3.1.2 Content

The schema for the Package Catalog is described in Annex E.

5.3.2 Electronic Device Description

5.3.2.1 Format

The EDD is an encoded file using the Electronic Device Description Language (EDDL) in accordance with 5.3.2.2.

The format of the EDD part is described in Table 3.

Table 3 – EDD part

Part	Content
Content Type	application/vnd.fdi.package.edd
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/edd
Filename	Not specified

5.3.2.2 Content

The EDDL is specified in IEC 61804. FDI specific rules for EDDs are described in Annex K.

5.3.3 User Interface Plug-in

5.3.3.1 Format

An FDI Package may contain one or more User Interface Plug-in parts. All User Interface Plug-in parts shall be identified by a package relationship.

The User Interface Plug-In represents a container element for UIP Variants (see 5.3.3.2.2.2), which contain the physical representation of a UIP (different variants for different platforms) that is consumed and executed by an FDI Client. A User Interface Plug-in shall at least provide one UIP Variant.

The variants are packaged into a single User Interface Plug-in using Open Packaging Convention as specified in 5.3.3.2. The User Interface Plug-in is not directly consumed by an FDI Server. The User Interface Plug-in is an optional part of different FDI Package types as specified in 4.3. The User Interface Plug-in is illustrated in Figure 12.

The UIP version information as well as the update and upgrade behavior shall follow the rules and concepts described in IEC 61804-1.

The default locale for UIPs and all contained variants is English (US). Optional language support is allowed according to market needs.

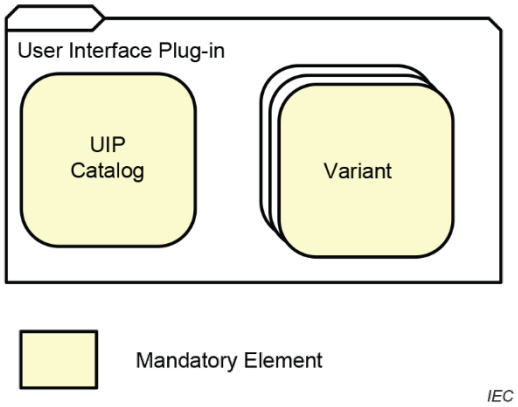

Figure 12 – User Interface Plug-in

The format for the User Interface Plug-in part is described in Table 4.

Table 4 – User Interface Plug-in part

Part	Content
Content Type	application/vnd.fdi.package.uip
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip
Filename	extension shall be .uip

5.3.3.2 Content

5.3.3.2.1 Use of Open Packaging Convention

5.3.3.2.1.1 Core

Core Properties are not used and shall be ignored by all FDI UIP Package consumers.

5.3.3.2.1.2 Thumbnails

Thumbnails are not used and shall be ignored by all FDI UIP Package consumers.

5.3.3.2.1.3 Digital Signatures

Digital signatures might be necessary for UIP Variants (see 5.3.3.2.2.2) depending on the implementation technology. Implementation technology details and necessary mechanisms to be implemented are described in IEC 62769-6.

The User Interface Plug-in as such however shall not be signed or applied signatures shall be ignored by the consuming FDI component.

5.3.3.2.2 User Interface Plug-in Parts

5.3.3.2.2.1 UIP Catalog

5.3.3.2.2.1.1 Format

The UIP Catalog is an XML file describing the properties of a UIP that are necessary to create the Information Model nodes in the FDI Server side and for deployment (FDI Server to FDI Client).

The UIP Catalog also hosts information about the physical starting element that has to be called/executed from a UIP Variant on the FDI Client. The format and type of this starting element are technology specific and therefore defined in IEC 62769-6.

The UIP Catalog shall be identified by a single package relationship.

The UIP Catalog part is an XML file whose schema is defined in Annex E. A User Interface Plug-in shall have only one UIP Catalog. The root element of the file is UipCatalog. The structure of the UipCatalog element is illustrated in Figure 13.

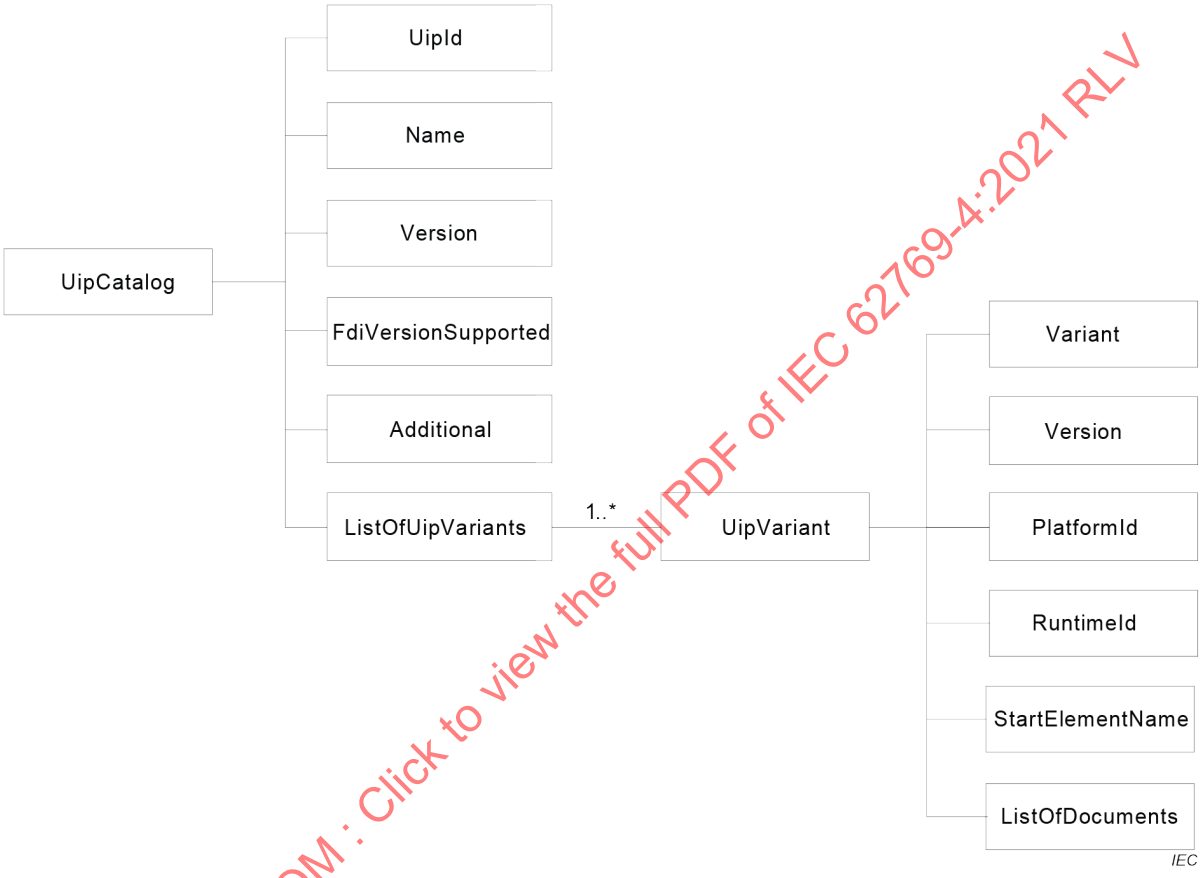

Figure 13 – UIP Catalog

The UIP Catalog part is described in Table 5.

Table 5 – UIP Catalog Part

Part	Content
Content Type	application/vnd.fdi.package.uip.catalog+xml
Root Namespace	http://fdi-cooperation.com/2010/uip-catalog
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip-catalog
Filename	uipcatalog.xml

5.3.3.2.2.1.2 Content

The schema for the UIP Catalog is described in Annex E.

5.3.3.2.2.2 UIP Variant

5.3.3.2.2.2.1 Format

A UIP Variant can be provided to support different versions of the base technology. It is a container for all physical files that represent the UIP Variant, independently of their type and number. The UIP Variant shall use the ZIP file format (compressed archive, application/zip media type) as bundling technology.

Use of the ZIP specification for bundling variants shall be constrained to the requirements in ISO/IEC 29500-2:2016, Annex C.

The UIP Variant part is described in Table 6.

Table 6 – UIP Variant Part

Part	Content
Content Type	application/zip
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/uip-variant
Filename	Not specified

5.3.3.2.2.2.2 Content

The content of the UIP Variant is specified in IEC 62769-6.

5.3.4 Attachments

5.3.4.1 Image

An FDI Package image attachment is a computer icon that represents the device type. Multiple images and resolutions of the device type are supported. Each image shall be PNG format and restricted to the following pixel resolutions: 256 × 256, 64 × 64, 32 × 32 or 16 × 16. The image part is described in Table 7.

If the Device Package is supposed to support handheld devices the provided image shall be available in a resolution of 16 pixels × 16 pixels.

Table 7 – Image Part

Part	Content
Content Type	image/png
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-image
Filename	Not specified

5.3.4.2 Documentation

Documentation parts include documentation such as product manuals and data sheets and shall be encoded in PDF or plain text format. The documentation Attachment is described in Table 8.

Table 8 – Documentation Part

Part	Content
Content Type	Application/pdf (specified in ISO 32000-1) Text/plain
Root Namespace	Not applicable
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-documentation
Filename	Not specified

5.3.4.3 Protocol Support File

The protocol support file part is a file not defined by this document but necessary to integrate the product into a system. Those files usually extend the integration mechanism provided in the basic part of the FDI Device Package (EDD, UIPs) by protocol-specific means.

Those files shall be neither executables nor binary files of any kind. In addition, those files shall not replace any mandatory elements of an FDI Device Package.

A list of protocol support files for the different communication protocols are defined in Annex F. The protocol-specific support file part is described in Table 9.

Table 9 – Protocol Support File Part

Part	Content
Content Type	Not specified here
Root Namespace	Not specified here
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Filename	Not specified here

5.3.4.4 FDI Registration Certificate

5.3.4.4.1 Format

The FDI Registration certificate is a single XML document per FDI Package, which can be present (optional FDI Package Element). In the scope of an FDI Package, it is a Package Element of the type Attachment.

Table 10 – FDI Registration Certificate Part

Part	Content
Content Type	application/vnd.fdi.package.registrationCert+xml
Root Namespace	Not specified here
Source Relationship	http://fdi-cooperation.com/2010/relationships/attachment-registrationCert
Filename	RegistrationCert.xml

5.3.4.4.2 Content

The FDI Registration Certificate is machine readable and is represented in XML Format and has the content and structure in accordance with Figure 14. The FDI Registration Certificate may be signed with a detached signature within the same document.

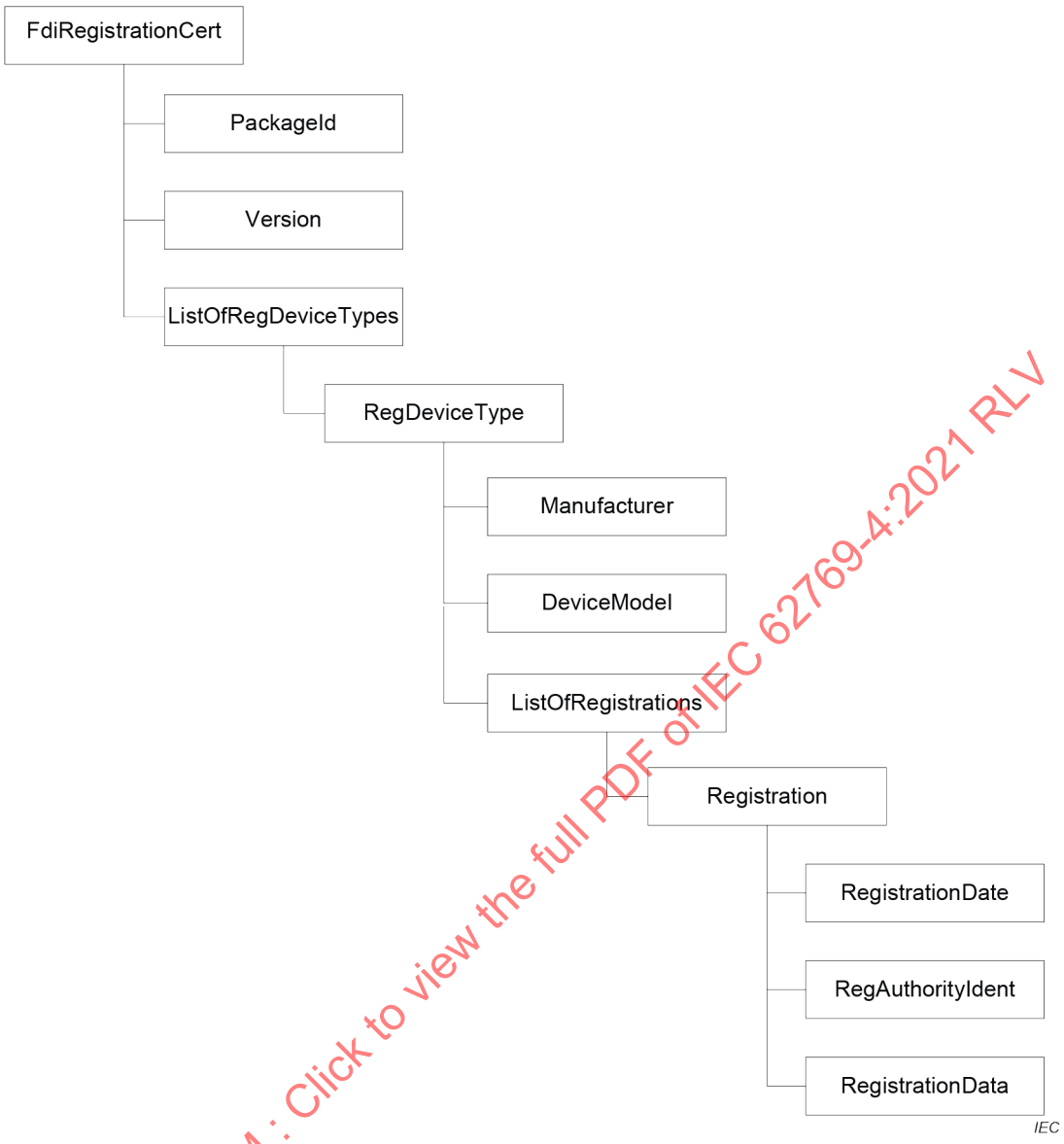

Figure 14 – FDI Registration Certificate

The schema for the FDI Registration Certificate is described in Annex E.

6 FDI Package Versioning

6.1 Version scheme

FDI elements use a major.minor.revision numeric versioning scheme for package and package elements. The initial version shall be 1.0.0.

Incompatible changes are indicated by incrementing the major number. Functional changes that still maintain compatibility to the major version are indicated by incrementing the minor number. Non-functional changes, such as editorial changes are indicated by incrementing the revision number. Rules for incrementing the version number are described in 6.3.

Examples of how to deploy different FDI Package types are described in Annex C.

6.2 Versioned elements

An FDI Package contains the version information that is described in Table 11.

Table 11 – Versioned Elements

Scope	Version Location	Schema Reference
FDI Package (see 5.3)	Package Catalog (see 5.3.1)	See Annex E, Version element of the Package complex type
UIP (see 5.3.3)	UIP Catalog (see 5.3.3.2.2.1)	See Annex E, Version element of the Uip complex type
UIP Variant (See 5.3.3.2.2.2)	UIP Catalog (see 5.3.3.2.2.1)	See Annex E, Version element of the UipVariant complex type

The following parts of the FDI Package have version mechanisms that are outside the scope of the FDI.

- EDD (see 5.3.2)
  - DD_REVISION and DEVICE_REVISION in accordance with IEC 61804-3
- Attachments (see 5.3.4)
  - Image (see 5.3.4.1)
  - Documentation (see 5.3.4.2)
  - Protocol Support File (see 5.3.4.3 and Annex F)

6.3 Version Hierarchy

The version change of FDI Package elements can influence the version of overlaying elements. Figure 15 illustrates those influences. The orange boxes show elements that are versioned using the FDI specific version mechanisms described in IEC 62769-1. The green boxes indicate elements that have version mechanisms that are not described by FDI. The white boxes describe elements that are not explicitly versioned.

All elements in the tree structure shown below influence the version of the parent elements in accordance with Table 12. Their version change causes a version change of the overlaying element.

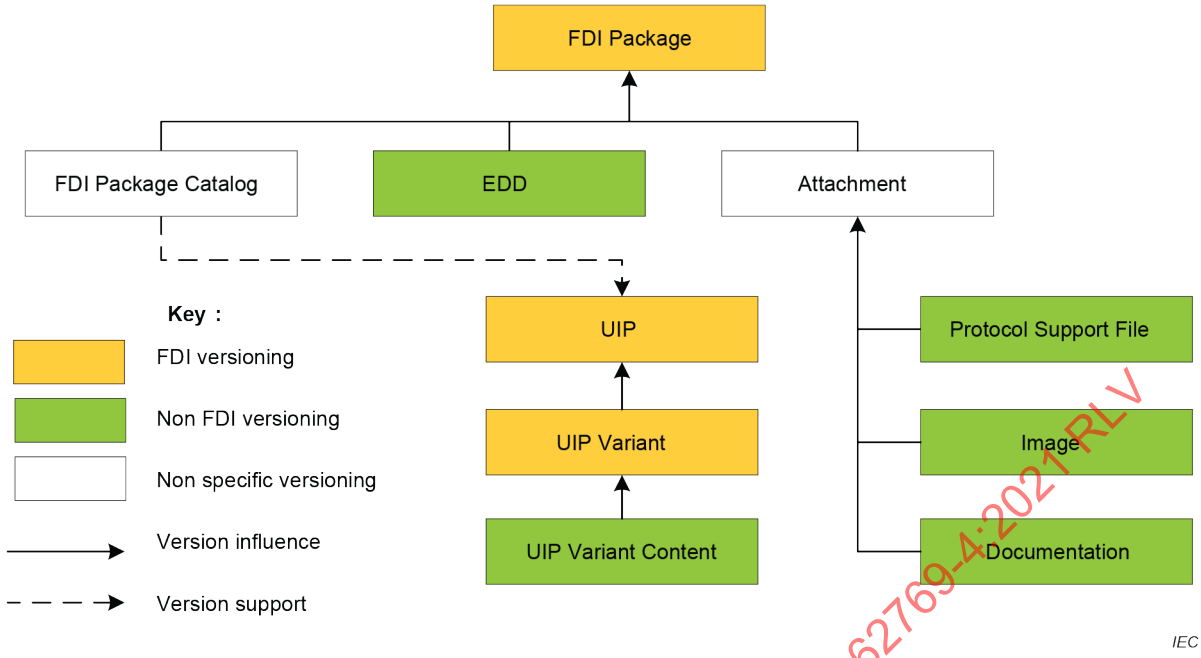

Figure 15 – Version Hierarchy

Table 12 describes a selection of possible changes to the different FDI Package elements and their direct influence on the FDI Package version. At a minimum, successive FDI Package versions shall increment the revision number.

Table 12 – Influence on FDI Package Version

Element	Version Level			FDI Package Type	
	Major	Minor	Revision	Device/ Profile/ Comm Package	UIP Package
Package Catalog	n/a	Modifying existing compatibility references. Change to ListOfSupportedUips (see Annex E). Only addition of wildcards is allowed for modifications to existing VersionSupported (see Annex E)	Editorial	x	x
EDD	Increment Device Revision	Increment DD_REVISION with same DEVICE_REVISION	n/a	x	
Attachments	Changes to Protocol Support Files	Changes to Protocol Support Files	Changes to Documentation, Images, Protocol Support Files	x	
UIP	Incompatible changes Technology driven	Functional enhancement Technology driven	Bug fix Technology driven	x	x

Annex G describes typical FDI Package life cycle use cases for a better understanding of the version hierarchy and the versioning concept.

6.4 UIP Compatibility

UIPs are delivered in an FDI Device Package, FDI Communication Package, FDI Profile Package or an FDI UIP Package. A UIP delivered in an FDI UIP Package shall not be delivered in any other package type. Therefore, an FDI Package may not be self-contained.

FDI Device Packages, FDI Communication Packages and FDI Profile Packages do not directly reference UIPs. Instead, the Package Catalog defines the SupportedUip (see Annex E) to define compatible UIPs. More than one installed UIP may be compatible with a given FDI Package version. This permits the FDI Package creators to provide bug fixes and functional enhancements to the FDI Package without the need to provide a major release of an FDI Package.

The version support described in the Package Catalog is defined as follows:

- The major release number shall be specified explicitly as a single number.
- The minor release number and revision number shall be specified explicitly as a single number or as a wildcard ('*'). If the minor release number is specified as a wildcard, then the revision number shall also be specified as a wildcard.

NOTE Examples of valid compatible version numbers are 1.3.1, 1.3.*, 1.*.*.

The use of a wildcard indicates that all minor numbers and/or revision numbers are compatible with the FDI Package. If the SupportedUip has been defined by using wildcards and there is more than one version of the UIP available, the FDI Server shall transfer the latest version to the FDI Client. System-specific implementations that allow coexistence of minor release or revisions releases regardless of version support wildcards are outside the scope of this document.

Figure 16 illustrates an example of selecting the most recent compatible UIP.

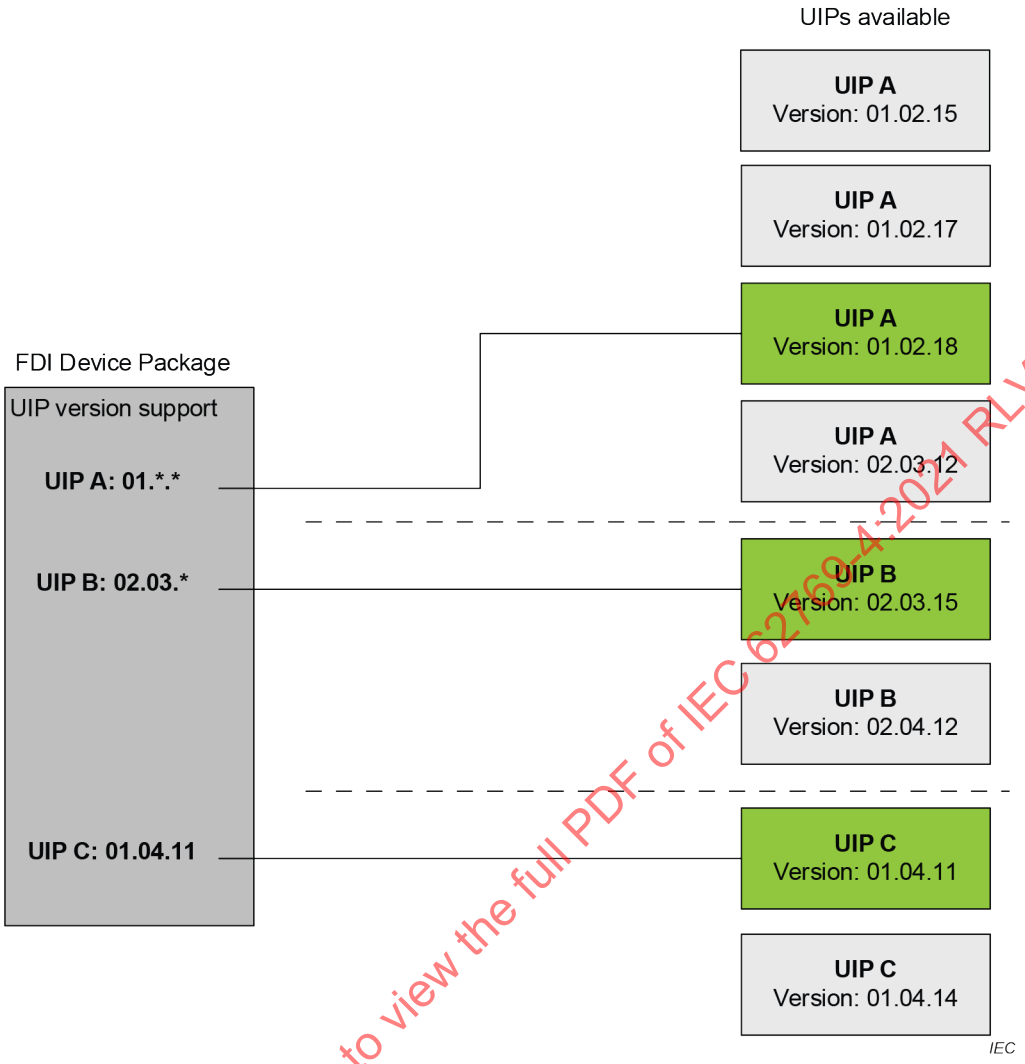

Figure 16 – UIP Version Support concept

7 Digital Signatures and Registration Certificates

7.1 Signed Elements and Certification documents

The FDI Package signing policy defines that the entire FDI Package shall be signed by the FDI Package originator (see 7.3). FDI Packages that have been registered by an FDI Registration Authority should contain an FDI Registration Certificate (special attachment, see 5.3.4.4), which shall be signed by an FDI Registration Authority (see 7.3). FDI Packages are registered if they are successfully conformance tested (see B.2.4).

Registered FDI Packages should carry one or more digitally signed FDI Registration Certificates to:

– indicate that the FDI Packages has been registered by an official FDI registration authority;

– indicate that the files in the package, which had been central for registration of the package, have not been altered after submission for registration.

FDI Packages as a whole (the surrounding entity covering all elements) shall be signed to:

– identify the originator (the signer) of the FDI Package;
– verify that the signed FDI Package hasn't been altered after the signature was applied.

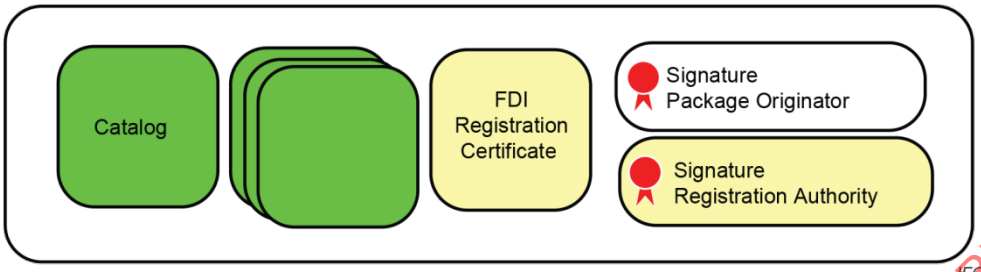

Figure 17 – FDI Package signing

The FDI Package originator first sends a package to the FDI Registration Authority. They perform defined conformance tests and additional tests according to their specific rules and test descriptions or agreements.

After successful testing, those Registration Authorities might issue an FDI Registration Certificate, which shall be signed by the individual FDI Registration Authority. The FDI Registration Certificate may contain more information about the registration, for example a hash on the parts of the FDI Package which had been covered by the conformance tests. The FDI Registration Certificate is incorporated into the FDI Package for release. After adding the FDI Registration Certificate to the FDI Package content, the package is again signed and afterwards released/published by the originator.

The FDI Package originator therefore takes over the responsibility that the FDI Registration Certificate is trustable by signing the entire FDI Package he releases.

7.2 Signing mechanism

All signatures within the FDI Package shall be made in accordance with the mechanism defined in ISO/IEC 29500-2. In addition to ISO/IEC 29500-2, the signature shall fulfill the following requirements:

– The information needed to validate the signature shall be part of the digital signature, i.e. the KeyInfo element specified in XML Signature Syntax and Processing is mandatory.
– Certificates used for signing shall be rooted in a Certificate Authority which is included in the trusted CAs of the Microsoft Windows Certificate Store[1].
– The algorithms used in creation of the signature (for hashing and encryption/decryption) shall be from the list of NIST recommended algorithms in FIPS 140-2, Annex A (NIST).
– The signature shall include a trusted timestamp in compliance with XAdES (XML Advanced Electronic Signatures – ETSI EN 319 132-1).
– Any signature shall include a CommitmentTypeIndication in accordance with ETSI TS 101 733. The used commitment types are specified in 7.3.

[1] Microsoft Windows Certificate Store is the trade name of a product supplied by Microsoft®. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

7.3 FDI Package Originator, FDI Registration Authority

The FDI Package Originator and the FDI Registration Authority have the following responsibilities:

– An FDI Package originator officially publishes an FDI Package and signs it to ensure the integrity of the FDI Package. The FDI Package can be created by a device vendor or a software solution provider. The publisher of an FDI Package may be a different person. The commitment type is ProofOfOrigin.

– An FDI Registration Authority has the right and the ability to perform FDI conformance tests on FDI Packages and to issue FDI Registration Certificates, Typically interest groups representing an FDI supported communication protocol or their authorized partners. The commitment type is ProofOfApproval.

7.4 FDI Host behavior

An FDI host system shall display a warning message when the FDI Package import procedure recognizes that:

– a digital signature on the package is not present or does not include all entities (files) inside the package,

– the digital signature as such is not trustable,

– the signature is broken, which indicates that the package has been modified after signing.

Additional security measures to be taken, if the warning message has been displayed, are in the responsibility of the FDI host system.

An FDI host system should display an information message showing which parts of the ones having gone into the registration have been changed when the FDI Package import procedure recognizes that:

– the unique identifier (PackageID) and the version (as defined in Annex E) of the FDI Package does not match the same information given as a part of the FDI Registration Certificate file;

– there is no FDI Registration Certificate present in the FDI Package;

– the included FDI Registration Certificate is not signed, the signature is not trustable, or the signature is broken.

An FDI host system can check the signature and certification status by reading the FDI Registration Certificate. A host shall provide a configuration, which allows to import an FDI Package, which does not include an FDI Registration Certificate. The functionality of this FDI Package shall not be limited.

Annex A
(normative)

File name conventions

A.1 Identification

Identification naming conventions shall be utilized to provide a unique way of identifying complete FDI Packages as well as elements of FDI Packages. Adherence to the identification rules will promote interoperability across systems. However, the names themselves shall not be the only mechanism for deployment.

Traditionally, machine-readable naming conventions have been utilized to uniquely link a file to a specific device and facilitate the import and use of device interface files, such as an EDD, making it difficult for users to determine if the required file was available in the file system. Because an FDI Package is the visible element to the user, a human-readable format is preferred over that of a machine-readable version.

A.2 FDI Package filename convention

Individual FDI Packages shall be identified by unique file names, which shall consist of the manufacture, the model or type, the revision and the protocol supported. Owing to filename persistence problems, the file name shall not be the only means to identify an FDI Package. Secure means of identifying an FDI Package shall be utilized to prevent inappropriate mixing of device to FDI Packages.

The FDI Package shall use the following naming convention

<manufacturer>.<description>.<major>.<minor>.<revision>.<protocol>.fdix

Each element of the filename is described in Table A.1.

The following rules for naming an FDI Package shall apply:

- All FDI Packages shall use the .fdix extension.
- Maximum name length including .fdix extension: 128 characters.
- The file name shall not include a space.

NOTE The maximum file name length is limited to 128 characters to reduce the probability of a path exceeding the maximum value.

Table A.1 – FDI Package Naming Convention

Filename component	Description
manufacturer	String representation of the manufacturer name of the device. The manufacturer shall not include a period.
description	A brief description of the package. The description shall not include a period. For an FDI Device Package, a string representation of the device type name.
major	Two-character numerical representation of the major release of the FDI Package.
minor	Two-character numerical representation of the minor release of the FDI Package.
revision	Two-character numerical representation of the revision of the FDI Package.
protocol	The communication profile family names are defined in the communication profiles (IEC 62769-1xx-x).

Annex B
(informative)

FDI Package Creation

B.1 General

Annex B describes the fundamentals of a possible FDI Package creation process by using standardized development tools and components available.

B.2 Tools and Components

B.2.1 Overview

FDI Packages consist of several different components – which also might have relationships and dependencies to each other – that are developed by using different implementation technologies and that follow different standards. This complex structure of FDI Packages requires a tool support to make an easy and economic development and maintenance possible.

B.2.2 FDI Reference Implementation/Common EDD Engine

The FDI Reference Implementation, including a common EDD Engine, ensures that a common implementation is used for an FDI Package implementation and test that has a defined behavior. The FDI Reference Implementation is part of both tools mentioned below.

B.2.3 FDI Package IDE

The FDI Package IDE provides everything that is necessary to manage development projects for the different kind of FDI Packages, the development of the descriptive parts, the linking of all remaining package parts, but also to do the actual packaging of the package.

B.2.4 FDI Device Package Conformance Test Tool

A developed FDI Package and especially FDI Device Packages are tested to prove the conformance of the implementation to the FDI Specification. This ensures interoperability. The conformance test is done by using the FDI Device Package Conformance Test Tool that executes defined test cases with the FDI Reference Implementation and the developed FDI Package.

B.3 Development

B.3.1 FDI Package core development

The FDI Package IDE allows the creation of a development project for the FDI Package, including the project type and version information. This development project can be used during the complete life cycle of the FDI Package. A development project wizard might be available to speed up the project creation process.

An editor component coming with the FDI Package IDE is available to implement the descriptive part (EDD) of the FDI Package. Features such as syntax checks, code folding, auto complete and wizards for complex constructs are available to support an efficient and safe development.

The FDI Package (depending on the FDI Package type) consists of several FDI Package parts. A project management component is also part of the FDI Package IDE, which allows the linkage of external FDI Package parts to a specific FDI Package development project. As soon as the developer has linked all FDI Package parts to the development project (and has also implemented EDD relationships if necessary), the FDI Package can be bound and packaged.

The FDI Package IDE generates parts of the Package Catalog depending on linked FDI Package parts, development project information and EDD source code. Some parts of the FDI Package, however, cannot be developed by using the FDI Package IDE. Examples are given in B.3.2 and B.3.3 and shown in Figure B.1.

B.3.2 User Interface Plug-in development

User Interface Plug-ins are developed using well known implementation technologies (for example, Microsoft.NET[2]). There are several powerful development tools available to use those implementation technologies (for example, Microsoft Visual Studio®[3]). Those tools should be used to implement the User Interface-Plug-ins needed. The ready implemented Plug-ins can then be imported into the FDI Package IDE development project to be referenced in the EDD and to be packaged into the FDI Package for release, but also to be tested under runtime conditions in conjunction with the descriptive part of the FDI Package. See also Figure B.1.

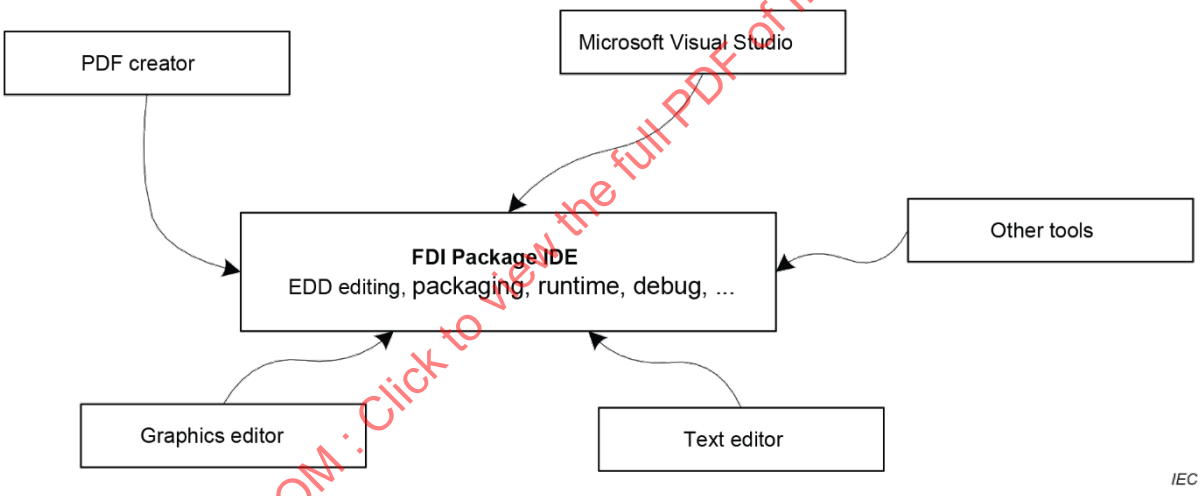

Figure B.1 – Tools used for FDI Package development

B.3.3 FDI Package Attachment development

There are different kinds of Attachments that can be incorporated into FDI Packages. Owing to the variety of possible and necessary Attachments, appropriate development tools depending on the type of Attachment shall be used to implement/create those. The Attachments can then be imported into the FDI Package IDE development project to be packaged into the FDI Package for release. See also Figure B.1.

---

2 Microsoft.NET is the trade name of a product supplied by Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

3 Microsoft Visual Studio is the trade name of a product supplied by Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

B.3.4 FDI Package binding and packaging

The last step of the FDI Package development is the packaging as such. In this development step, all developed and related parts of the FDI Package are packed according to this specification. Checks are performed to ensure consistency. The output can then be used for conformance testing and in systems.

B.3.5 Conformance Test

The use of the FDI Package Conformance Test tool which is also part of the FDI Package IDE to ensure the conformance of the FDI Package with the FDI specification marks the last step before releasing the product. The Conformance Test tool uses specified test cases with an FDI runtime engine to check the single features of the FDI that have or may be implemented into an FDI product (an FDI Package in this case).

Annex C
(informative)

FDI Package deployment

C.1 General

Annex C describes sequence examples of how to deploy different FDI Package types to different system architectures. The sequence and detailed features are system-specific.

For FDI Servers, no conditional deployment of UIP Variants is defined because package content can be consumed by different kinds of FDI Clients.

For standalone FDI host systems, FDI Server and FDI Client application are a single integrated entity. A standalone FDI host system can perform conditional deployment of the UIP Variants in accordance with the integrated FDI Client capabilities.

C.2 Scenarios

C.2.1 FDI Package deployment to PC based client/server systems

C.2.1.1 FDI Device Packages/FDI Profile Packages/FDI Communication Packages

The following steps apply for the deployment of FDI Device Packages, FDI Profile Packages and FDI Communication Packages to an FDI Server.

a) The user chooses an FDI Package from the file system.
b) The system validates the FDI Package signature and integrity.
c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies
   - that the PackageType shall be "Device", "Profile", or "Communication";
   - that the FDIVersionSupported shall be equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server;
   - the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by the FDI.
d) The system adds the FDI Package information to the system specific device catalog.
e) The system reads all UIPs stored in the FDI Package and adds all UIPs and all available variants to the system specific UIP catalog.
f) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.
g) The system stores the entire FDI Device Package content.
h) The system reads the EDD for each device type and creates Information Model (IM) type nodes.

C.2.1.2 FDI UIP Packages

The following steps apply for the deployment of UIP Packages to an FDI Server.

a) The user chooses an FDI Package from the file system.
b) The system validates the FDI Package signature and integrity.
c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies
   - that the PackageType is "Uip";

- that the FDIVersionSupported is equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and
- the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by the FDI.

d) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.

e) The system stores the entire FDI Device Package content.

f) The system reads all UIPs stored in the FDI Package and adds all UIPs and all available variants to the system specific UIP catalog.

C.2.2 FDI Package deployment to an FDI standalone system

C.2.2.1 FDI Device Packages/FDI Profile Packages/FDI Communication Packages

The following steps apply for the deployment of FDI Device Packages, FDI Profile Packages and FDI Communication Packages to an FDI standalone system.

a) The user chooses an FDI Package from the file system.

b) The system validates the FDI Package signature and integrity.

c) The system reads the FDI Package Catalog root element as defined in 4.2.1 and verifies

- that the PackageType is "Device", "Profile", or "Communication";
- that the FDIVersionSupported is equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and
- the version and PackageId against already installed versions of this device type and handles the update and upgrade accordingly. The deployment is aborted if there is a higher version installed since downgrades are not supported by FDI.

d) The system reads all UIP Variants for all UIPs in the FDI Package and verifies

- that the PlatformId and RuntimeId is supported by the integrated FDI Client; and
- the version and UipId against already imported UIP Variants and handles the update and upgrade accordingly. If there is already a higher version installed the import is aborted since downgrades are not supported by the FDI.

e) The system adds the UIP information of matching and imported UIPs to the system specific UIP catalog.

f) The system reads the ListOfSupportedUips for each device type and notifies the user if a required UIP is not installed.

g) The system stores the relevant FDI Device Package content.

h) The system reads the EDD for each device type and creates Information Model (IM) type nodes.

C.2.2.2 FDI UIP Packages

The following steps apply for the deployment of UIP Packages to an FDI standalone system.

a) The user chooses an FDI Package from file system.

b) The system validates the FDI Package signature and integrity.

c) The system reads the FDI Package Catalog information, as defined in 4.2.1:

- the PackageType attribute is being checked (see Annex E) for "Uip"
- the FDIVersionSupported shall be equal to the Major version (depending on the use of wildcards instead of actual version numbers, the Minor and the Revision part of the version information shall also be interpreted) of the FDI Server; and

- the system imports UIPs including UIP Variants and checks for already imported versions of the particular UIP and handles the update and upgrade accordingly. If there is already a higher version installed, the import is aborted since downgrades are not supported by the FDI.

d) The system reads all UIP Variants for all UIPs in the FDI Package and verifies:
   - that the PlatformId and RuntimeId is supported by the integrated FDI Client; and
   - the version and UipId against already imported UIP Variants and handles the update and upgrade accordingly. If there is already a higher version installed, the import is aborted since downgrades are not supported by FDI.

e) The system stores the relevant FDI Device Package content.

Annex D
(informative)

Example

D.1 General

The purpose of Annex D is to provide an overview of ISO/IEC 29500-2 and an example FDI Device Package implementation. It is not intended to provide all the details that might be necessary for the implementation of an FDI Package.

Unless otherwise stated in Annex D, the initialism OPC refers to Open Packaging Conventions.

D.2 Open Packaging Conventions

D.2.1 Overview

The Open Packaging Conventions (OPC) are a container-file technology specified in ISO/IEC 29500-2. OPC-based documents are ZIP archives that contain XML, binary and other types of files. They combine the advantages of maintaining the independence and the integrity of the files embedded in the document while having a single integrated package.

An OPC package consists of parts and relationships as shown in Figure D.1. Parts refer to the content being packaged such as binary and text files. Relationships define associations between the package, parts and external resources.

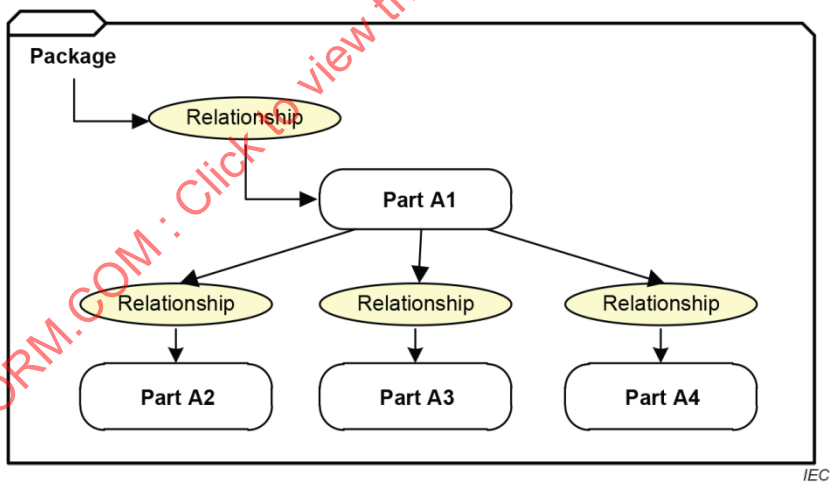

Figure D.1 – Parts and relationships in a package

D.2.2 Parts

OPC packages can store parts that contain any type of data (text, images, XML, binary, whatever). Parts can be organized as a hierarchy of folders that is similar to a file system. The OPC specification allows any folder organization that is convenient for the application.

Every part in a package has a unique URI-compliant part name along with a specified content-type expressed in the form of a MIME media type. The MIME media types for all the parts stored in the package are defined in an XML file named "/[Content_Types].xml".

In an FDI Package, the parts include the Package Catalog, EDD, UIPs and Attachments.

D.2.3 Relationships

In addition to a hierarchy of folders and parts, OPC allows the definition of relationships among parts. Relationships provide a layer of indirection, so dependencies are not hardcoded into the content.

Relationships allow problem domain and application specific relationships to be defined and enforced by systems and tools. By navigating through the network of associations, one can have random access to related content.

Relationships are composed of four elements:

- an identifier (ID);
- an optional source (the package or a part within the package);
- a relationship type (a URI-style expression that defines the type of the relationship);
- a target (a URI to another part within the package or to an external resource).

The relationships are stored in XML files with the extension ".rels" within subfolders named "/_rels".

In an FDI Package, the relationships are used to express the association of the package to the Package Catalog, the EDD, the UIP and the Attachments.

The following clause, for instance, defines the relationship identified as "rIdUip1", which establishes the association between the Package and the Package Catalog.

```
  <Relationship Id="rIdUip1"
    Type="http://fdi-cooperation.com/2010/relationships/package-
catalog"
    Target="fdipackage/catalog.xml"/>
```

D.2.4 OPC Core Features

D.2.4.1 Overview

The content types file /[Content_Types].xml, the package relationships file /_rels/.rels, and the part relationships files in the subfolders _rels are the core parts of an OPC package.

The filename [Content_Types].xml in any folder, the subfolder name _rels, and the file extension .rels within such directory are the only three reserved names for files stored in an OPC package.

D.2.4.2 Content Types

The file /[Content_Types].xml defines the MIME media types for all the parts stored in the package. It defines default mappings based on file extensions, along with overrides for specific parts with content types that are different from the file extension defaults.

Table D.1 and Table D.2 show examples of standard MIME media types and examples of FDI custom MIME media types that can be used in FDI packages.

Table D.1 – Examples of standard MIME media types that can be used in FDI packages

Type of content	Standard MIME media type
XML file	application/xml
PNG image	image/png
PDF document	application/pdf
OPC relationships	application/vnd.openxmlformats-package.relationships+xml

Table D.2 – Examples of FDI custom MIME media types that can be used in FDI Packages

Type of content	FDI custom MIME media type
Package catalog	application/vnd.fdi.package.catalog+xml
EDD	application/vnd.fdi.package.edd
UIP	application/vnd.fdi.package.uip

The following clause in the /[Content_Types].xml file defines the default MIME media type for any XML files in the package.

```
<Default Extension="xml" ContentType="application/xml"/>
```

The following clause in the same file will override the definition above for the package catalog part only:

```
<Override PartName="/fdipackage/catalog.xml"
    ContentType="application/vnd.fdi.package.catalog+xml"/>
```

D.2.4.3 Package Relationships

The root level /_rels folder stores the relationships for the package as a whole. The /_rels folder normally contains a file named .rels, an XML file where the starting package-level relationships are stored.

D.2.4.4 Part Relationships

Each part may have its own relationships. If the part has relationships, they will be stored in an XML file within the _rels folder that is a sibling of that part. That file takes the part name with a .rels appended to it.

Because the Package Catalog part has relationships to other package parts, there is a file named catalog.xml.rels inside the /_rels folder that defines those relationships.

D.2.5 OPC Additional features

D.2.5.1 Core Properties

Core properties consist of package metadata. They enable users to get and set well-known and common sets of property metadata within packages, such as categorization of the content, status of package (e.g. draft, reviewed, final), date of creation, identification of the creator, keywords, and language. The core properties are not used in FDI Packages (see 5.2).

D.2.5.2 Thumbnails

Thumbnails are images that are used as graphical representation of parts of a package or a package as a whole. The use of thumbnails in an FDI Package is optional (see 5.2).

D.2.5.3 Digital Signatures

Digital signatures can be used to enable consumers to validate the integrity of the contents. The use of digital signatures is mandatory in FDI Packages (see 5.2).

D.3 Creation and Handling of FDI Packages

As long as the conventions are followed, OPC files can be created, opened and modified just as any ordinary ZIP file by using standard ZIP file tools. However, there exists higher level support to handle them so that developers do not have to cope with all the peculiarities of OPC. OPC is natively supported in Microsoft .NET Framework 3.0. Open source libraries also exist for other languages. Ultimately, FDI-specific tools are expected to provide automated support for the creation and handling of FDI Packages.

Given the catalog information, the EDD, the UIPs and attachments for a specific device type, a hypothetical FDI packaging tool is capable of creating the FDI package for that device type, as depicted in Figure D.2.

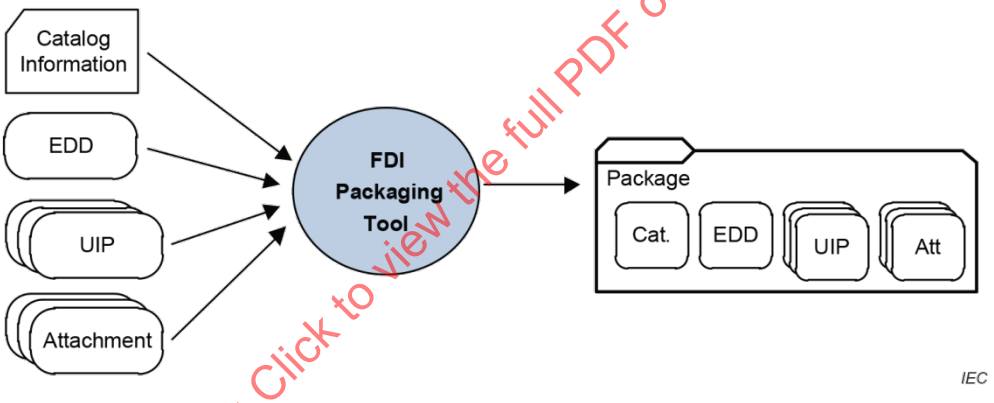

Figure D.2 – Creating an FDI Package with the content files

D.4 FDI Device Package Example

D.4.1 Overview

Figure D.3 represents an example FDI Device Package. The root directory of the FDI Device Package contains three directories and one file. The directory names fdipackage and uip are examples and are not defined by this document. The _rels directory is used to store the package relationships and is defined by ISO/IEC 29500-2.

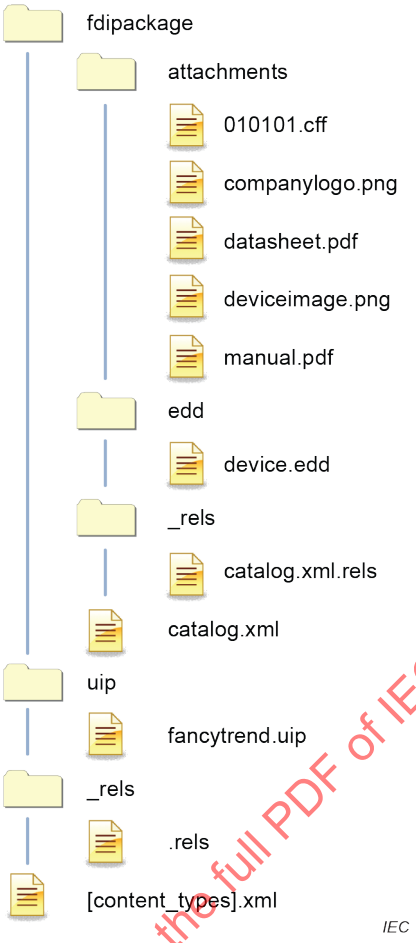

Figure D.3 – FDI Device Package Example

The file [Content_Types].xml is defined by ISO/IEC 29500-2 and defines the content type of the parts in the package. It is required for all packages. All parts in the package shall have a content type identified by this file.

An example for /[Content_Types].xml is listed below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
    <Default Extension="rels"
      ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
    <Default Extension="xml" ContentType="application/xml"/>
    <Default Extension="uip" ContentType="application/vnd.fdi.package.uip"/>
    <Default Extension="png" ContentType="image/png"/>
    <Default Extension="pdf" ContentType="application/pdf"/>
    <Default Extension="edd" ContentType="application/vnd.fdi.package.edd"/>
    <Default Extension="cff" ContentType="application/vnd.ff.cff"/>
    <Override PartName="/fdipackage/catalog.xml"
      ContentType="application/vnd.fdi.package.catalog+xml"/>
</Types>
```

In this example, the Package Catalog is identified by the part /fdipackage/catalog.xml with content type application/vnd.fdi.package.catalog+xml in the /[Content_Types].xml. The device has a protocol-specific CFF file. It is the responsibility of the protocol organization to define the content type associated with that file type. The content type for the CFF is an example and is not currently specified by the Fieldbus Foundation.

The /rels/.rels part defines the package relationships per ISO/IEC 29500-2.

An example for /rels/.rels is listed below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Relationships
  xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId1"
      Type="http://fdi-cooperation.com/2010/relationships/package-catalog"
      Target="fdipackage/catalog.xml"/>
    <Relationship
      Id="rIdUip1" Type="http://fdi-cooperation.com/2010/relationships/uip"
      Target="uip/fancytrend.uip"/>
</Relationships>
```

In accordance with this document, the Package Catalog shall be identified by a single package relationship. The FDI Server identifies the Package Catalog part by retrieving the part associated with this standard relationship type. In this example, the catalog relationship is represented by

```
<Relationship Id="rId1"
  Type="http://fdi-cooperation.com/2010/relationships/package-catalog"
  Target="fdipackage/catalog.xml"/>
```

The value for relationship id is not specified by this standard. ISO/IEC 29500-2 requires that each relationship have a unique relationship ID.

An FDI Device Package may also provide one or more UIPs. In accordance with this document, a UIP shall be identified by a package relationship. The FDI Server identifies a UIP in a package by searching for all parts with the associated relationship type. In this example, a single UIP is identified by

```
<Relationship Id="rIdUip1" Type="http://fdi-cooperation.com/2010/relationships/uip"
  Target="uip/fancytrend.uip"/>
```

An example for /fdicatalog/catalog.xml is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:Catalog
xmlns:fdi="http://fdi-cooperation.com/2010/package"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>ef377fd0-5de5-11df-a08a-0800200c9a66</PackageId>
  <PackageType>Device</PackageType>
  <Version>1.0.0</Version>
  <FdiVersionSupported>1.0.0</FdiVersionSupported>
  <ManufacturerName>ACME Transmitters</ManufacturerName>
  <ManufacturerContact>42 Wallaby Way, Sydney,
Australia</ManufacturerContact>
  <ManufacturerUrl>http://acme.local</ManufacturerUrl>
  <ManufacturerImage>rIdMfrLogo</ManufacturerImage>
  <ListOfDeviceTypes>
    <DeviceType>
      <Name>
        <value>Temperature Transmitter</value>
        <value xml:lang="fr">Transmetteur de température</value>
        <value xml:lang="de">Temperatur-Transmitter</value>
      </Name>
      <ClassificationId>SENSOR_TEMPERATURE</ClassificationId>
      <ListOfInterfaces>
        <Interface>
          <ListOfCommunicationProfiles>
            <CommunicationProfile>foundation_h1</CommunicationProfile>
          </ListOfCommunicationProfiles>
          <Version>5.0.0</Version>
          <Manufacturer>0xff00</Manufacturer>
```

```
          <DeviceModel>0x1234</DeviceModel>
          <CommunicationRole>CLIENT</CommunicationRole>
          <ListOfCommunicationProfileSupportFiles>

<CommunicationProfileSupportFile>rIdCFF</CommunicationProfileSupportFi
le>
          </ListOfCommunicationProfileSupportFiles>
        </Interface>
      </ListOfInterfaces>
      <Edd>rIDEDD</Edd>
      <ListOfSupportedDeviceRevisions>
        <DeviceRevision>1.0.0</DeviceRevision>
      </ListOfSupportedDeviceRevisions>
      <ListOfImages>
        <Image>rIdPicture1</Image>
      </ListOfImages>
      <ListOfDocuments>
        <Document>rIdDocument1</Document>
        <Document>rIdDocument2</Document>
      </ListOfDocuments>
      <ListOfSupportedUips>
        <SupportedUip>
          <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
          <Name>Fancy Trend</Name>
          <Version>1.1.*</Version>
         <Optional>true</Optional>
        </SupportedUip>
      </ListOfSupportedUips>
    </DeviceType>
  </ListOfDeviceTypes>
</fdi:Catalog>
```

The Package Catalog part will reference other package parts through relationship IDs. Those relationship IDs will be defined in a relationship file name by the part and appended with the .rels suffix. In this example, this part is named fdipackage/_rels/catalog.xml.rels.

An example for /fdipackage/_rels/catalog.xml.rels is listed below. The relationship IDs are not defined by this document. In accordance with ISO/IEC 29500-2, these relationship IDs shall be unique. The ID names in this example were selected to better illustrate referencing.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/device.edd" Id="rIdEDD"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-image"
      Target="attachments/deviceimage.png" Id="rIdPicture1"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
      Target="attachments/datasheet.pdf" Id="rIdDocument1"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
      Target="attachments/manual.pdf" Id="rIdDocument2"/>
    <Relationship
      Type="http://fdi-cooperation.com/2010/relationships/attachment-protocol"
      Target="attachments/010101.cff" Id="rIdCFF"/>
</Relationships>
```

In this example, the product documentation is identified by the following.

```
<ListOfDocuments>
    <Document>rIdDocument1</Document>
    <Document>rIdDocument2</Document>
</ListOfDocuments>
```

The catalog identifies the files by the relationship ids that are found in the corresponding /fdipackage/_rels/catalog.xml.rels.

```
<Relationship
  Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
  Target="attachments/datasheet.pdf" Id="rIdDocument1"/>
<Relationship
  Type="http://fdi-cooperation.com/2010/relationships/attachment-documentation"
  Target="attachments/manual.pdf" Id="rIdDocument2"/>
```

The example FDI Device Package identifies one supported UIP.

```
<SupportedUip>
    <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
    <Name>Fancy Trend</Name>
    <Version>1.1.*</Version>
</SupportedUip>
```

This document permits UIPs to be delivered in the FDI Device Package, or the UIPs may be delivered in a separate FDI UIP Package. In this example, the UIP is delivered in the FDI Device Package.

UIPs are also encoded using ISO/IEC 29500-2, but they are not considered a valid FDI Package type. An FDI server will not directly consume a UIP. A UIP shall be encoded in a valid FDI Package as shown in this example.

D.4.2 User Interface Plug-in

The following example is for the UIP referenced from the example FDI Package in Clause D.4. This UIP has two variants, one targeted for the workstation and the other for a mobile environment. The UIP part is encoded according to ISO/IEC 29500-2. Figure D.4 shows the structure of the example UIP. In this example, the filename for the UIP is fancytrend.uip, and is explicitly referenced in the package relationship ID in Clause D.4.

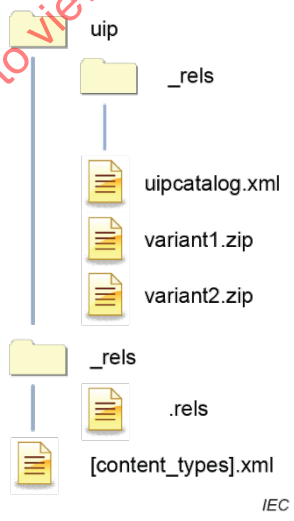

Figure D.4 – User Interface Plug-in Example (fancytrend.uip)

Similar to the example in Clause D.4, a package conforming to ISO/IEC 29500-2 shall have a /[content_types].xml and _rels/.rels part. In this example, the directory name uip is only an example and not specified by this document.

An example for [Content_Types].xml is listed below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
    <Default Extension="rels"
      ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
    <Default Extension="xml" ContentType="application/xml"/>
    <Default Extension="zip" ContentType="application/zip"/>
    <Override PartName="/uip/uipcatalog.xml"
      ContentType="application/vnd.fdi.package.uip.catalog+xml"/>
</Types>
```

An example for _rels/.rels part is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId1"
      Type="http://fdi-cooperation.com/2010/relationships/uip-catalog"
      Target="uip/uipcatalog.xml"/>
</Relationships>
```

In accordance with this document, the UIP Catalog part shall be identified by a package relationship. This is represented by the following code of the _rels/.rels part.

```
<Relationship Id="rId1"
  Type="http://fdi-cooperation.com/2010/relationships/uip-catalog"
  Target="uip/uipcatalog.xml"/>
```

The relationship ID is only an example. The relationship type allows an FDI Server to identify the part corresponding to the UIP Catalog. In this example, the catalog is the /uip/uipcatalog.xml part.

An example for /uip/uipcatalog.xml list listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:UipCatalog  xmlns:cat="http://fdi-cooperation.com/2010/package"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UipId>f67e4ad0-5de5-11df-a08a-0800200c9a66</UipId>
    <Name>Fancy Trend</Name>
    <Version>1.0.0</Version>
    <FdiVersionSupported>1.0.0</FdiVersionSupported>
    <Additional>Best trend ever</Additional>
    <ListOfUIPVariants>
        <UIPVariant>
            <Variant>rIDVariant1</Variant>
            <Version>1.0.0</Version>
            <PlatformId>Workstation</PlatformId>
            <RuntimeId>String</RuntimeId>
            <StartElementName>Variant1.assembly</StartElementName>
        </UIPVariant>
        <UIPVariant>
            <Variant>rIDVariant2</Variant>
            <Version>1.0.0</Version>
            <PlatformId>Mobile</PlatformId>
            <RuntimeId>String</RuntimeId>
            <StartElementName>Variant2.assembly</StartElementName>
        </UIPVariant>
    </ListOfUIPVariants>
</fdi:UipCatalog>
```

The UIP Catalog will have a corresponding relationship part. In this example, the part is /uip/_rels/uipcatalog.xml.rels.

An example for /uip/_rels/uipcatalog.xml.rels is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/uip-variant"
      Target="variant1.zip" Id="rIdVariant1"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/uip-variant"
      Target="variant2.zip" Id="rIdVariant2"/>
</Relationships>
```

In this example, the UIP Variants are stored in the /uip directory. It is possible that these could be stored in a different location. The target of the relationship would be updated to properly identify the UIP Variant.

D.4.3 EDD reference to UIP

The following EDD fragment can be used to identify the UIP in this example. The following code includes only the necessary attributes to establish the relationship. Other mandatory attributes have been removed for clarity.

```
MENU mymenu
{
    ITEMS
    {
        pFrancyTrend
    }
}

PLUGIN pFrancyTrend
{
    UUID f67e4ad0-5de5-11df-a08a-0800200c9a66;
}
```

D.4.4 FDI Registration Certificate

The example shows an FDI Device Package that describes device model 1234. The first registration was issued by FDI Registration Corp., Singapore in December 2010. A second registration was issued by FDI Registration Corp., Cologne in March 2011.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:FdiRegistrationCert
  xmlns:fdi=http://fdi-cooperation.com/2010/package
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>ef377fd0-5de5-11df-a08a-0800200c9a66</PackageId>
  <Version>1.3.4</Version>
  <ListOfRegDeviceTypes>
    <RegDeviceType>
      <Manufacturer>00ff00</Manufacturer>
      <DeviceModel>1234</DeviceModel>
      <ListOfRegistrations>
        <Registration>
          <RegistrationDate>2010-12-31</RegistrationDate>
          <RegAuthorityIdent>FDI Registration Corp.,
Singapore</RegAuthorityIdent>
          <RegistrationData>
            <value>Test Tool Version 2.1, Conformance Test<value>
            <value xml:lang="de">Test Tool Version 2.1, Conformance
Test</value>
          </RegistrationData>
        </Registration>
        <Registration>
          <RegistrationDate>2011-03-07</RegistrationDate>
```

```
          <RegAuthorityIdent>FDI Registration Corp.,
Köln</RegAuthorityIdent>
          <RegistrationData>
            <value>Test Tool Version 3.2, Conformance Test<value>
            <value xml:lang="de">Test Tool Version 3.2, Conformance
Test</value>
          </RegistrationData>
        </Registration>
      </ListOfRegistrations>
    </RegDeviceType>
  </ListOfRegDeviceTypes>
</fdi:FdiRegistrationCert>
```

The FDI Registration Certificate format does not only allow the description of several device types, but also the history listing of the registrations of a single type.

Annex E
(normative)

Schema

E.1 Target Namespace

The target namespace defined for the catalog document is defined by:

```
<xs:schema
   xmlns:fdi="http://fdi-cooperation.com/2010/package"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://fdi-cooperation.com/2010/package"
   elementFormDefault="unqualified" version="0.14.0">
```

E.2 Catalog

The Catalog element is the mandatory root element for the Package Catalog of an FDI Package.

The XML schema for a Catalog element is:

```
<xs:element name="Catalog" type="fdi:PackageT"/>
```

E.3 ClassificationIdT

The ClassificationIdT simple type specifies the device type classification (e.g. for sorted representation of installed device types represented by FDI Device Packages).

The XML schema for a ClassificationIdT is an enumeration and matches the discrete values of the CLASSIFICATION attribute specified in IEC 61804-3.

E.4 CommunicationProfileT

The CommunicationProfileT simple type specifies a unique identifier for the communication profile family and protocol using the pattern family_protocol. Valid values are specified in communication profiles (IEC 62769-1xx-x).

The XML schema for a CommunicationProfileT string type is:

```
<xs:simpleType name="CommunicationProfileT">
   <xs:restriction base="xs:string"/>
</xs:simpleType>
```

E.5 CommunicationRoleT

The CommunicationRoleT simple type specifies the supported communication function to differentiate if the device is a communication server, a gateway, or simple device type.

The XML schema for a CommunicationRoleT enumeration type is:

```
<xs:simpleType name="CommunicationRoleT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="SERVER"/>
      <xs:enumeration value="CLIENT"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a CommunicationRoleT enumeration type are described in Table E.1.

Table E.1 – Enumerations of CommunicationRoleT

Enumeration	Description
SERVER	Provides communication services for a specified protocol
CLIENT	Uses communication services implemented for a specified protocol

E.6 CommunicationServerT

The CommunicationServerT complex type specifies information to identify an FDI Communication Server.

The XML schema for a CommunicationServerT type is:

```
<xs:complexType name="CommunicationServerT">
  <xs:sequence>
    <xs:element name="ProductUri" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

The elements of a CommunicationServerT type are described in Table E.2.

Table E.2 – Elements of CommunicationServerT

Element	Description
ProductUri	ProductURI of the FDI Communication Sever

E.7 DeviceTypeT

The DeviceTypeT complex type specifies a device type definition.

The XML schema for a DeviceTypeT type is:

```
<xs:complexType name="DeviceTypeT">
   <xs:sequence>
      <xs:element name="Name" type="fdi:ListOfLocalizedStringsT"/>
      <xs:element name="ClassificationId"
         type="fdi:ClassificationIdT"/>
      <xs:element name="ListOfInterfaces"
         type="fdi:ListOfInterfacesT"/>
      <xs:element name="Edd" type="fdi:RelationshipIdT"/>
      <xs:element name="ListOfSupportedDeviceRevisions"
         type="fdi:ListOfSupportedDeviceRevisionsT" minOccurs="0"/>
      <xs:element name="ListOfImages" type="fdi:ListOfDeviceImagesT"
         minOccurs="0"/>
      <xs:element name="ListOfDocuments" type="fdi:ListOfDocumentsT"
         minOccurs="0"/>
      <xs:element name="ListOfSupportedUips"
         type="fdi:ListOfSupportedUipsT" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a DeviceTypeT type are described in Table E.3.

Table E.3 – Elements of DeviceTypeT

Element	Description
Name	Name of the device type The name can be localized
ClassificationId	Classification of the device type
ListOfInterfaces	List of interfaces supported by the device type
Edd	Reference to the EDD of the device type within the FDI Package
ListOfSupportedDeviceRevisions	List of compatible revisions of the device type that is described in this FDI Package
ListOfImages	List of references to images for this device type
ListOfDocuments	List of references to documents for this device type
ListOfSupportedUips	List of supported UIPs that are referenced by the EDD of this device type

E.8 FdiRegistrationCert

The FdiRegistrationCert element is the mandatory root element of the registration certificate.

The XML schema for a FdiRegistrationCert element is:

```
<xs:element name="FdiRegistrationCert"
   type="fdi:FdiRegistrationCertT"/>
```

E.9 FdiRegistrationCertT

The FdiRegistrationCertT complex type specifies the details of the registration certificate.

The XML schema for a FdiRegistrationCertT type is:

```
<xs:complexType name="FdiRegistrationCertT">
   <xs:sequence>
      <xs:element name="PackageId" type="fdi:UuidT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="ListOfRegDeviceTypes"
         type="fdi:ListOfRegDeviceTypesT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a FdiRegistrationCertT type are described in Table E.4.

Table E.4 – Elements of FdiRegistrationCertT

Element	Description
PackageId	A unique identifier for the FDI Package
Version	Package version
ListOfRegDeviceTypes	List of one registered device type

E.10 InterfaceT

The InterfaceT complex type specifies elements of the interface supported by the device type.

The XML schema for an InterfaceT type is:

```
<xs:complexType name="InterfaceT">
   <xs:sequence>
      <xs:element name="ListOfCommunicationProfiles"
         type="fdi:ListOfCommunicationProfilesT"/>
      <xs:element name="Version">
         <xs:simpleType>
            <xs:restriction base="fdi:VersionT"/>
         </xs:simpleType>
      </xs:element>
      <xs:element name="Manufacturer" type="xs:string" minOccurs="0"/>
      <xs:element name="DeviceModel" type="xs:string" minOccurs="0"/>
      <xs:element name="CommunicationRole"
         type="fdi:CommunicationRoleT"/>
      <xs:element name="ListOfCommunicationProfileSupportFiles"
         type="fdi:ListOfProtocolSupportFilesT" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of an InterfaceT type are described in Table E.5.

Table E.5 – Elements of InterfaceT

Element	Description
ListOfCommunicationProfiles	List of communication profiles supported by the interface
Version	Version of the communication profile
Manufacturer	Manufacturer identifier as specified in the communication profile (IEC 62769-1xx-x) Shall be omitted in case of CommunicationRole = SERVER and shall be provided in case of CommunicationRole = CLIENT
DeviceModel	Device type identifier as specified in the communication profile (IEC 62769-1xx-x) Shall be omitted in case of CommunicationRole = SERVER and shall be provided in case of CommunicationRole = CLIENT
CommunicationRole	Supported communication functions for a specified protocol. A Communication Server shall describe at least one Interface element that contains CommunicationRole SERVER. A Communication Server shall not describe an Interface with CommunicationRole CLIENT. A Gateway shall describe at least one Interface element that contains CommunicationRole SERVER. A Gateway shall describe one Interface element that contains CommunicationRole CLIENT. A Device shall describe one Interface element that contains CommunicationRole CLIENT. A Device shall not describe an Interface with CommunicationRole SERVER.
ListOfCommunicationProfileSupportFiles	Optional list of communication profile support files

E.11 ListOfCommunicationProfilesT

The ListOfCommunicationProfilesT complex type is a list of one or more CommunicationProfiles.

The XML schema for a ListOfCommunicationProfilesT type is:

```
<xs:complexType name="ListOfCommunicationProfilesT">
   <xs:sequence>
      <xs:element name="CommunicationProfile"
         type="fdi:CommunicationProfileT" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfCommunicationProfilesT type are described in Table E.6.

Table E.6 – Elements of ListOfCommunicationProfilesT

Element	Description
CommunicationProfile	Unique identifier for the communication profile. Valid values are listed in the communication profiles (IEC 62769-1xx)

E.12 ListOfDeviceImagesT

The ListOfDeviceImagesT complex type is a list of one or more images.

The XML schema for a ListOfDeviceImagesT type is:

```
<xs:complexType name="ListOfDeviceImagesT">
   <xs:sequence >
      <xs:element name="Image" type="fdi:RelationshipIdT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDeviceImagesT type are described in Table E.7.

Table E.7 – Elements of ListOfDeviceImagesT

Element	Description
Image	Reference to an image in the FDI Package

E.13 ListOfDeviceTypesT

The ListOfDeviceTypesT complex type is a list of one device type.

The XML schema for a ListOfDeviceTypesT type is:

```
<xs:complexType name="ListOfDeviceTypesT">
   <xs:sequence >
      <xs:element name="DeviceType" type="fdi:DeviceTypeT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDeviceTypesT type are described in Table E.8.

Table E.8 – Elements of ListOfDeviceTypesT

Element	Description
DeviceType	Device type definition

E.14 ListOfDocumentsT

The ListOfDocumentsT complex type is a list of one or more documents.

The XML schema for a ListOfDocumentsT type is:

```
<xs:complexType name="ListOfDocumentsT">
   <xs:sequence>
      <xs:element name="Document" type="fdi:RelationshipIdT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfDocumentsT type are described in Table E.9.

Table E.9 – Elements of ListOfDocumentsT

Element	Description
Document	Reference to a document in the FDI Package

E.15 ListOfInterfacesT

The ListOfInterfacesT complex type is a list of one or more interfaces supported by the device type.

The XML schema for a ListOfInterfacesT type is:

```
<xs:complexType name="ListOfInterfacesT">
   <xs:sequence>
      <xs:element name="Interface" type="fdi:InterfaceT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfInterfacesT type are described in Table E.10.

Table E.10 – Elements of ListOfInterfacesT

Element	Description
Interface	Interface definition

E.16 ListOfLocalizedStringsT

The ListOfLocalizedStringsT complex type is a list of one or more strings localized by attributes.

The XML schema for a ListOfLocalizedStringsT type is:

```
<xs:complexType name="ListOfLocalizedStringsT">
   <xs:sequence>
      <xs:element name="value" type="fdi:LocalizedStringT"
         minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>

</xs:complexType>
```

The elements of a ListOfLocalizedStringsT type are described in Table E.11.

Table E.11 – Elements of ListOfLocalizedStringsT

Element	Description
value	Value for a localized string

E.17 ListOfProtocolSupportFilesT

The ListOfProtocolSupportFilesT complex type is a list of one or more protocol support files.

The XML schema for a ListOfProtocolSupportFilesT type is:

```
<xs:complexType name="ListOfProtocolSupportFilesT">
   <xs:sequence>
      <xs:element name="CommunicationProfileSupportFile"
         type="fdi:RelationshipIdT" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfProtocolSupportFilesT type are described in Table E.12.

Table E.12 – Elements of ListOfProtocolSupportFilesT

Element	Description
CommunicationProfileSupportFile	Reference to a communication profile support file in the package

E.18 ListOfRegDeviceTypesT

The ListOfRegDeviceTypesT complex type is a list of one registered device type.

The XML schema for a ListOfRegDeviceTypesT type is:

```
<xs:complexType name="ListOfRegDeviceTypesT">
   <xs:sequence>
      <xs:element name="RegDeviceType" type="fdi:RegDeviceTypeT"
         minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfRegDeviceTypesT type are described in Table E.13.

Table E.13 – Elements of ListOfRegDeviceTypesT

Element	Description
RegDeviceType	A registered device type

E.19 ListOfRegistrationsT

The ListOfRegistrationsT complex type is a list of one or more registered device types.

The XML schema for a ListOfRegistrationsT type is:

```
<xs:complexType name="ListOfRegistrationsT">
   <xs:sequence>
      <xs:element name="Registration" type="fdi:RegistrationT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfRegistrationsT type are described in Table E.14.

Table E.14 – Elements of ListOfRegistrationsT

Element	Description
Registration	Registration describing registration details as per the registration authority

E.20 ListOfSupportedDeviceRevisionsT

The ListOfSupportedDeviceRevisionsT complex type is a list of one or more device revisions that are compatible with this FDI Package.

The XML schema for a ListOfSupportedDeviceRevisionsT type is:

```
<xs:complexType name="ListOfSupportedDeviceRevisionsT">
   <xs:sequence>
      <xs:element name="DeviceRevision" type="fdi:VersionSupportedT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfSupportedDeviceRevisionsT type are described in Table E.15.

Table E.15 – Elements of ListOfSupportedDeviceRevisionsT

Element	Description
DeviceRevision	Revision of the device that is compatible with this device type in the FDI Package

E.21 ListOfSupportedUipsT

The ListOfSupportedUipsT complex type is a list of one or more User Interface Plug-ins.

The XML schema for a ListOfSupportedUipsT type is:

```
<xs:complexType name="ListOfSupportedUipsT">
   <xs:sequence>
      <xs:element name="SupportedUip" type="fdi:SupportedUipT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfSupportedUipsT type are described in Table E.16.

Table E.16 – Elements of ListOfSupportedUipsT

Element	Description
SupportedUip	Description of a UIP that is compatible with this device type

E.22 ListOfUipVariantsT

The ListOfUipVariantsT complex type is a list of one or more UIP variants.

The XML schema for a ListOfUipVariantsT type is:

```
<xs:complexType name="ListOfUipVariantsT">
   <xs:sequence>
      <xs:element name="UIPVariant" type="fdi:UipVariantT"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a ListOfUipVariantsT type are described in Table E.17.

Table E.17 – Elements of ListOfUipVariantsT

Element	Description
UIPVariant	UIP Variant definition

E.23 LocalizedStringT

The LocalizedStringT complex type specifies localized string specified by the attribute. Strings with no language specification default to English.

The XML schema for a LocalizedStringT type is:

```
<xs:complexType name="LocalizedStringT">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="xml:lang" use="optional" default="en"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

The attributes of a LocalizedStringT type are described in Table E.18.

Table E.18 – Attributes of LocalizedStringT

Attribute	Description
lang	Language code according to ISO 639-1

E.24 PackageT

The PackageT complex type specifies the elements of the Package Catalog.

The XML schema for a PackageT type is:

```
<xs:complexType name="PackageT">
   <xs:sequence>
      <xs:element name="PackageId" type="fdi:UuidT"/>
      <xs:element name="PackageType" type="fdi:PackageTypeT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="FdiVersionSupported" type="fdi:VersionT"/>
      <xs:element name="ManufacturerName" type="fdi:String256T"/>
      <xs:element name="ManufacturerContact"
         type="fdi:LocalizedStringT" minOccurs="0"/>
      <xs:element name="ManufacturerUrl" type="xs:anyURI"
         minOccurs="0"/>
      <xs:element name="ManufacturerImage" type="fdi:RelationshipIdT"
         minOccurs="0"/>
     <xs:element name="CommunicationServer"
        type="fdi:CommunicationServerT" minOccurs="0"/>
      <xs:element name="ListOfDeviceTypes"
         type="fdi:ListOfDeviceTypesT" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a PackageT type are described in Table E.19.

Table E.19 – Elements of PackageT

Element	Description
PackageId	Unique identifier of the FDI Package
PackageType	Identifies the unique type of the FDI Package
Version	Unique version number for the FDI Package according to the FDI version convention
FdiVersionSupported	FDI Technology Version supported by the FDI Package
ManufacturerName	Manufacturer name of the FDI Package
ManufacturerContact	General contact information for the manufacturer of the FDI Package. Contact information can be localized
ManufacturerUrl	Website contact for the manufacturer of the FDI Package
ManufacturerImage	Reference to an image in the FDI package of the manufacturer's logo. The image shall be PNG format and have a resolution of 256 × 256
CommunicationServer	Information on how to discover an FDI Communication Server This information shall only be provided for packages of an FDI Communication Server.
ListOfDeviceTypes	List of device types available in the FDI Package. Mandatory for PackageTypes Device, Profile and Communication

E.25 PackageTypeT

The PackageTypeT simple type specifies the FDI Package type.

The XML schema for a PackageTypeT enumeration type is:

```
<xs:simpleType name="PackageTypeT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Device"/>
      <xs:enumeration value="Uip"/>
      <xs:enumeration value="Communication"/>
      <xs:enumeration value="Profile"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a PackageTypeT enumeration type are described in Table E.20.

Table E.20 – Enumerations of PackageTypeT

Enumeration	Description
Device	FDI Device Package
Uip	FDI UIP Package
Communication	FDI Communication Package
Profile	FDI Profile Package

E.26 PlatformT

The PlatformT simple type defines the target platform for the UIP Variant.

The XML schema for a PlatformT enumeration type is:

```
<xs:simpleType name="PlatformT">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Workstation"/>
      <xs:enumeration value="Mobile"/>
   </xs:restriction>
</xs:simpleType>
```

The enumeration values of a PlatformT enumeration type are described in Table E.21.

Table E.21 – Enumerations of PlatformT

Enumeration	Description
Workstation	Workstation platform
Mobile	Mobile platform

E.27 RegDeviceTypeT

The RegDeviceTypeT complex type specifies a device type definition.

The XML schema for a RegDeviceTypeT type is:

```
<xs:complexType name="RegDeviceTypeT">
   <xs:sequence>
      <xs:element name="Manufacturer" type="xs:string"/>
      <xs:element name="DeviceModel" type="xs:string"/>
      <xs:element name="ListOfRegistrations"
         type="fdi:ListOfRegistrationsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a RegDeviceTypeT type are described in Table E.22.

Table E.22 – Elements of RegDeviceTypeT

Element	Description
Manufacturer	Specific manufacturer identifier given by the communication profile rules
DeviceModel	Specific device type identifier given by the communication profile rules
ListOfRegistrations	Collection of Registrations with detailed registration information

E.28 RegistrationT

The RegistrationT complex type specifies details about the registration.

The XML schema for a RegistrationT type is:

```
<xs:complexType name="RegistrationT">
   <xs:sequence>
      <xs:element name="RegistrationDate" type="xs:date"/>
      <xs:element name="RegAuthorityIdent" type="xs:string"/>
      <xs:element name="RegistrationData"
         type="fdi:ListOfLocalizedStringsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a RegistrationT type are described in Table E.23.

Table E.23 – Elements of RegistrationT

Element	Description
RegistrationDate	Date when Registration Certificate was issued in format YYYY-MM-DD
RegAuthorityIdent	Identification information of the FDI Registration Authority (typically name and address)
RegistrationData	Data indicating the scope of the registration (description of test procedures, versions, validity, ...) – localized

E.29 RelationshipIdT

The RelationshipId simple type specifies the relationship ID in a part's relationship item for an embedded item within the FDI Package as specified in ISO/IEC 29500-2:2011, 9.3.2.

The XML schema for a RelationshipIdT type is:

```
<xs:simpleType name="RelationshipIdT">
   <xs:restriction base="xs:ID"/>
</xs:simpleType>
```

E.30 String256T

The String256T simple type specifies a string restricted to 256 or fewer characters.

The XML schema for a String256T type is:

```
<xs:simpleType name="String256T">
   <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
   </xs:restriction>
</xs:simpleType>
```

E.31 SupportedUipT

The SupportedUipT complex type specifies elements of a User Interface Plug-in.

The XML schema for a SupportedUipT type is:

```
<xs:complexType name="SupportedUipT">
   <xs:sequence>
      <xs:element name="UipId" type="fdi:UuidT"/>
      <xs:element name="Name" type="fdi:String256T"/>
      <xs:element name="Version" type="fdi:VersionSupportedT"/>
      <xs:element name="Optional" type="xs:boolean"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a SupportedUipT type are described in Table E.24.

Table E.24 – Elements of SupportedUipT

Element	Description
UipId	Identifier for a UIP
Name	Name of the UIP
Version	Versions of the UIP that are compatible with this FDI Package
Optional	If true, then the UIP is optional for the proper device integration. If false, then the UIP is required for proper device integration.

E.32 UipCatalog

The UipCatalog is the mandatory root element for the UIP Catalog of a User Interface Plug-in.

The XML schema for a UipCatalog element is:

```
<xs:element name="UipCatalog" type="fdi:UipT"/>
```

E.33 UipStyleT

The UipStyleT simple type specifies the style a UIP should run.

The XML schema for a UipStyleT enumeration type is:

```
<xs:simpleType name="UipStyleT" default="DIALOG">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WINDOW"/>
    <xs:enumeration value="DIALOG"/>
  </xs:restriction>
</xs:simpleType>
```

The enumeration values of a UipStyleT enumeration type are described in Table E.25.

Table E.25 – Enumerations of UipStyleT

Enumeration	Description
WINDOW	UIP should run as a modeless window If the parent starting the UIP is running modal, the UIP will run modal, otherwise modeless
DIALOG	UIP will always run as a modal window

E.34 UipT

The UipT complex type specifies the elements of the User Interface Plug-in catalog.

The XML schema for a UipT type is:

```
<xs:complexType name="UipT">
   <xs:sequence>
      <xs:element name="UipId" type="fdi:UuidT"/>
      <xs:element name="Name" type="fdi:String256T"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="FdiVersionSupported" type="fdi:VersionT"/>
      <xs:element name="Additional" type="fdi:String256T"/>
      <xs:element name="Style" type="fdi:UipStyleT "/>
      <xs:element name="ListOfUipVariants"
         type="fdi:ListOfUipVariantsT"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a UipT type are described in Table E.26.

Table E.26 – Elements of UipT

Element	Description
UipId	Unique identifier of the UIP
Name	Name of the UIP
Version	Version of the UIP
FdiVersionSupported	Version of the FDI Technology supported by this UIP
Additional	Additional information about the UIP
Style	Style of the UIP identifying where running modal or non-modal
ListOfUIPVariants	List of UIP Variants provided by this UIP

E.35 UipVariantT

The UipVariantT complex Type specifies the elements of a User Interface Plug-in variant.

The XML schema for a UipVariantT type is:

```
<xs:complexType name="UipVariantT">
   <xs:sequence>
      <xs:element name="Variant" type="fdi:RelationshipIdT"/>
      <xs:element name="Version" type="fdi:VersionT"/>
      <xs:element name="PlatformId" type="fdi:PlatformT"/>
      <xs:element name="RuntimeId" type="xs:string"/>
      <xs:element name="CpuInformation" type="xs:string"
         minOccurs="0">
      <xs:element name="StartElementName" type="xs:string"/>
      <xs:element name="ListOfDocuments" type="fdi:ListOfDocumentsT"
         minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The elements of a UipVariantT type are described in Table E.27.

Table E.27 – Elements of UipVariantT

Element	Description
Variant	Reference to the UIP variant within the FDI Package
Version	Version of the UIP Variant
PlatformId	Platform of the UIP Variant
RuntimeId	Runtime environment of the UIP as specified in IEC 62769-6
CpuInformation	The element value provides additional information about the execution environment associated with the UIP runtime. The allowed values are specified in IEC 62769-6
StartElementName	Element that is loaded on an FDI Client to start the UIP Variant as specified in IEC 62769-6
ListOfDocuments	Optional list of references to documents for this UIP Variant

E.36 UuidT

The UuidT simple type specifies a universally unique identifier as specified by ISO/IEC 11578. The UuidT is restricted to the formal xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx where x is a hexadecimal digit.

The XML schema for a UuidT type is:

```
<xs:simpleType name="UuidT">
   <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
         [0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
   </xs:restriction>
</xs:simpleType>
```

E.37 VersionSupportedT

The VersionSupportedT simple type specifies version compatibility. Version can be explicitly referenced (e.g. 1.1.1) or the * wild card can be used to specify a range of compatibility (e.g. 1.1.*). Wildcards are permitted for minor release and revision.

The XML schema for a VersionSupportedT type is:

```
<xs:simpleType name="VersionSupportedT">
   <xs:restriction base="xs:string">
      <xs:maxLength value="16"/>
      <xs:pattern value="(\d+)\.((\d+\.(\d+|\*))|(\*\.\*))"/>
   </xs:restriction>
</xs:simpleType>
```

E.38 VersionT

The VersionT simple type specifies a version restricted to the format major.minor.revision.

The XML schema for a VersionT type is:

```
<xs:simpleType name="VersionT">
   <xs:restriction base="xs:string">
      <xs:maxLength value="16"/>
      <xs:pattern value="(\d+)\.(\d+)\.(\d+)"/>
   </xs:restriction>
</xs:simpleType>
```

Annex F
(normative)

Communication protocol specific profiles

Communication protocol specifics are described in so-called "Protocol-specific profile documents".

The creation and maintenance of those documents is the responsibility of the respective interest group representing the communication protocol (see Table F.1).

Table F.1 – Communication protocol interest groups (alphabetical order)

Protocol	Interest Group
FOUNDATION Fieldbus[a)]	FieldComm Group
HART[b)], WirelessHART[c)]	FieldComm Group
PROFIBUS[d)], PROFINET[e)]	PROFIBUS and PROFINET International
ISA100 [f)]	International Society of Automation (ISA)

a) In accordance with IEC 61784-1, CPF1.

b) In accordance with IEC 61784-1, CP9/1.

c) In accordance with IEC 62591.

d) In accordance with IEC 61784-1, CPF3.

e) In accordance with IEC 61784-2, CPF3.

f) In accordance with IEC 62734.

NOTE See the trade name declarations in IEC 61784-1 and IEC 61784-2.

Annex G
(informative)

FDI Package life cycle use cases

G.1 New Device Type

Adding new devices to a plant is a typical use case when a plant or part of a plant is being extended.

The new device type may be handled by an FDI Package that is already installed or it may need a new FDI Package that represents the new device type in the system. An already installed FDI Package might also be upgraded or a new specific FDI Package is required in order to support a new device type.

G.2 Replacement of Device

During the life cycle of a plant, it may happen that there is a need to repair or replace a device or device module. For example, in the case of Modular Devices, individual parts of the device are replaceable. Table G.1 shows device replacement guidelines.

Table G.1 – Device Replacement Guidelines

Device Manufacturer	Scenario	Guideline
Same manufacturer	Same type and same device version	—
Same manufacturer	Same type and new device version	May require an FDI Package update or upgrade
Same manufacturer	New type that is functionally compatible	May require an FDI Package update or upgrade or a new FDI Package
Same manufacturer	New type, that is functionally incompatible	Similar to that of the guideline for the new device type
Different manufacturer	Any	Similar to that of the guideline for the new device type

The device vendor shall communicate which FDI Package versions can be used for the new device revision. The vendor shall provide information on how to detect if the FDI Package shall be updated/upgraded and where to get this update/upgrade. Furthermore, the vendor shall provide a new or updated FDI Package version, if a device revision (Hardware) is incompatible with an already existing FDI Package version.

G.3 Firmware enhancements

An update of the device firmware may be required to fix errors that were detected in the firmware during the life cycle of the device. The device vendor provides maintenance of the device firmware by firmware updates, spare parts or replacement devices. Updating the device firmware is equivalent to the replacement of a device by the same type but new version (updated firmware).

The functionality of a device may be extended by upgrading the device firmware or changing the configuration of the device, for example, by installing a new device module. Table G.2 shows firmware enhancement guidelines.

Table G.2 – Firmware enhancement guidelines

Firmware	Guideline
Update	An update of firmware versions shall not require a new FDI Package version
Upgrade	In order to use this additional or enhanced functionality, an FDI Package upgrade may be required. Nevertheless the existing FDI Package version should be able to work with the new device firmware without leveraging the new device functionality

G.4 FDI Package life cycle polices

An FDI Package (including updates/upgrades) shall be shipped with product documentation including installation requirements, installation guideline, product description and release notes.

G.5 FDI Package update

A newer version of an FDI Package shall support all device types and data of the previous version of the FDI Package.

An FDI Package update should not change the scope of use. This includes:

- the available functionality (also user interface related);
- the supported device types;
- the supported bus protocols;
- the data items;
- the supported operating systems.

G.6 FDI Package upgrade

A newer version of an FDI Package shall support all device types and data of the previous version of the FDI Package.

An FDI Package upgrade might change the scope of use. This includes:

- the additional available functionality (also user interface related);
- the additional supported device types;
- the additional supported bus protocols;
- the additional data items;
- the additional supported operating systems.

G.7 FDI Package replacement/exchange

A system should implement an FDI Package replacement strategy in order to support replacement of an FDI Package by another FDI Package in a project. The system should provide a means to detect if an already installed FDI Package can handle the new device type or if a new FDI Package shall be installed.

G.8 FDI Package uninstallation

The FDI Package shall provide all means for an easy removal. An unsuccessful removal shall be indicated by the system.

Commonly used components shall be handled correctly (software parts and components shall be removed only, if no other references from other software programs exist). Existing FDI Package-specific files shall not be deleted automatically and shall be reusable in FDI Packages provided by the same vendor.

If an FDI Package, which is used and instantiated in a system, has been removed for any reason, the system shall be able to indicate which FDI Package is missing. It shall inform the user about removed FDI Packages and its supported device types: vendor, device name, type and version. The Device Instance data of the removed FDI Package shall not be deleted from the Information Model until the instances are removed by the user.

Annex H
(normative)

Health Status Method

H.1 Background

Many devices contain embedded intelligence to calculate diagnostic conditions. Other devices may have limited embedded processing and rely on external business logic processing to calculate device diagnostic conditions. Diagnostic data representation may be in various forms and may be influenced by the device communication profile.

H.2 Device Health Status model

The health status state provides a high level, consistent structured view to the current operating condition of a device independent of device or communication profiles. The health status state is calculated in an EDD method by accessing one or more device variables, calculating the health status state and returning a standard value to the application.

Some devices may offer configuration capability to map specific device diagnostic information to the health status state. The configuration of conditions to the health status state is device or communication profile-specific and is not part of this document.

The health status state shall be calculated in accordance with Table H.1. In the event of multiple conditions, the state with the lowest priority shall be returned.

Table H.1 – Health Status State

Health Status State	Priority	Conditions
Indeterminate	0	The health status is unavailable and therefore indeterminate. For example, the device may not be connected, a communication fault has occurred, or the device does not support the health status state.
Failure	1	Output signal is invalid due to malfunction in the field device or its peripherals.
Function Check	2	Output signal is temporarily invalid (e.g. frozen) due to ongoing work on the device.
Out of Specifications	3	Deviations from the permissible ambient or process conditions determined by the device itself through self-monitoring or faults in the device itself indicate that the measuring uncertainty of sensors or deviations from the set value in actuators is probably greater than expected under operating conditions.
Maintenance Required	4	Although the output signal is valid, the wear reserve is nearly exhausted, or a function will soon be restricted due to operational conditions.
Good	5	The device is operating under typical operating conditions such that Maintenance Requirement, Out of Specification, Failure and Function Check are not active.

H.3 Standard EDD Method signature

The EDD shall implement the GetHealthStatus method to provide access to the health status state. The method definition will be specific to the EDD. The method definition can use communication Builtins and shall not use user interface Builtins. See IEC 61804-4:2020, 5.3 for a list of communication Builtins and user interface Builtins.

The GetHealthStatus method shall return the health status state priority value in accordance with Table H.1. Devices that do not support calculating the health status state shall return 0.

```
METHOD GetHealthStatus
{
   LABEL "GetHealthStatus";
   TYPE unsigned char;
   DEFINITION
   {
      /* device specific definition */

      /* return health status priority */
   }
}
```

For modular, block-oriented devices, multiple health statuses may be available. In this case, the method name shall use the prefix GetHealthStatus_ (e.g. METHOD GetHealthStatus_TB). Block-oriented health status methods shall be listed in the METHOD_ITEMS attribute of the associated BLOCK_A declaration.

H.4 Performance considerations

Accessing health status information via a standard EDD method requires business logic processing in the FDI Server. The method will typically require at least one communication access to the device to collect the health status. Continuous scanning of the health status across several device and device networks may have a serious impact on the performance of the underlying communication networks.

Underlying communication networks may provide optimized methods (e.g. asynchronous event driven messages) for obtaining health status information for continuous condition-based monitoring.

Annex I
(normative)

Modular devices

I.1 Concept

The concept of modular devices is shown in Figure I.1 and is as follows:

1) The entire modular device is described in a single package.
2) The device's modular structure and related configuration rules are described in a single EDD file. This EDD file represents the top-level topology element of the modular device's structure. This EDD file is referred in the catalog schema.
3) EDD files describing the modules are contained in separate EDD files, which are not exposed in the catalog XML. The reference to these modules' EDD files is made from the COMPONENT defined attribute named EDD.
4) Packaging of other package elements as it is defined in 4.2 is not touched.

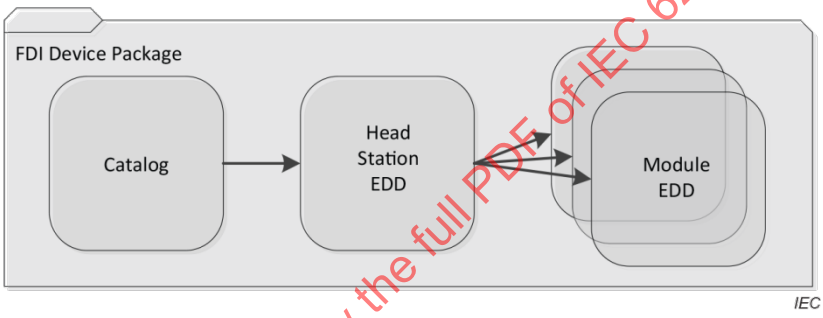

Figure I.1 – Modular device's package

I.2 EDDL usage profile

FDI Packages describing a modular device shall use the following EDDL defined constructs to describe the modular device's structure (topology) and related configuration rules:

1) COMPONENT
2) COMPONENT_FOLDER
3) COMPONENT_RELATION

The following EDDL defined syntax elements shall not be used:

1) COMPONENT_REFERENCE
2) INTERFACE
3) REQUIRED_INTERFACE
4) SUPPLIED_INTERFACE
5) FILTER

NOTE The rationale behind this decision is to reduce complexity for the FDI host implementation and for FDI Package creation. The restriction also protects the integrity of modular device description of one vendor since FDI does not support the extension of an existing modular device description with externally (other vendor) defined modules. This could happen if FDI supports using the EDDL defined syntax element COMPONENT_REFERENCE.

I.3 Processing recommendations

I.3.1 Monolithic device with device variants

This example shows how a pressure transmitter may be applied in different applications and for different measurement ranges.

The user places the top-level element in the topology. Now the host application can ask the user which device variant shall be used. (The same information can be read from the device based on the device vendor implemented "DETECT" function.) In order to define the actually needed device variant, the host application shall read the EDD and determine all COMPONENT and COMPONENT_FOLDER declarations.

I.3.2 Remote IOs

The user places the top-level element in the topology, which is the Remote IOs Head Station. Device variants have to be selected in accordance with the procedure described in I.3.1.

For the purpose of the configuration of the module setup, the host needs to read the EDD and determines all COMPONENT, COMPONENT_FOLDER and COMPONENT_RELATION declarations to understand the device internal module catalog and the related configuration rules. The host can cache the device internal module catalog that is used only for the module configuration of this particular Head Station.

If Head Stations variants are described in separate EDDs, all these EDDs shall be referred to in the Catalog XML. These Head stations can share a common set of modules. The device internal module catalog shall be described in all Head stations EDDs. (This can be solved using "#include" in EDD source code).

I.3.3 How to identify the top-level topology element

All topology elements of the modular device are based on COMPONENT or COMPONENT_FOLDER declarations. The following text describes how an FDI host can find the topmost topology element inside an EDD file.

The FDI host has to find all COMPONENT declarations that do not use the EDD attribute. These COMPONENT declarations belong to internal hierarchy of the Head Station. The top-level declaration of this hierarchy can either be a COMPONENT or a COMPONENT_FOLDER. This top-level declaration corresponds to the device type described in the Catalog XML.

I.3.4 Packaging details example

Based on the description found in D.2.4 and D.4.1, I.3.4 provides additional information that helps to understand how the module EDD files need to be added beside the Head Station EDD file, which is also referred in the Catalog.XML file. The following example shows how three EDD files are integrated in a single package. There is one EDD file for the Head Station (Target="edd/HeadStation.edd") and two module EDD files (Target="edd/Module_A.edd" and Target="edd/Module_B.edd").

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/HeadStation.edd" Id="rIdEDD_HeadStation"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/Module_A.edd" Id="rIdEDD_Module_A"/>
    <Relationship Type="http://fdi-cooperation.com/2010/relationships/edd"
      Target="edd/Module_B.edd" Id="rIdEDD_Module_B"/>
    …
</Relationships>
```

The entire set of EDD files can be found based on the specified relation type (Type = "http://fdi-cooperation.com/2010/relationships/edd").

The following catalog example is an excerpt to emphasize the concept of how the EDD file references work. The value calalog.xml defined element <EDD> refers to the package defined relation identifier (rIdEDD_HeadStation) that enables to retrieve the actual EDD file.

```
<DeviceType>
      <Name>
        <value>Modular remote IO</value>
       . . .
      </Name>
      <ClassificationId>REMOTEIO</ClassificationId>
      . . .
      <Edd>rIdEDD_HeadStation</Edd>
      . . .
</DeviceType>
```

Annex J
(normative)

FDI Communication Packages for FDI Communication Server

J.1 General

Details on packages for the different profiles are defined in Annex F. Annex J defines details on FDI Communication Packages used for the description and reference of FDI Communication Servers. They can be considered independent of technology profiles. This only considers the package, not the FDI Communication Server itself, which is defined in more detail in IEC 62769-7.

J.2 Protocol Support File

No additional file is required for FDI Communication Server packages.

J.3 CommunicationProfile definition

No values of CommunicationProfile are defined for FDI Communication Server packages.

J.4 Profile Device

There is no concept of a profile device for an FDI Communication Server.

J.5 Protocol version information

There is no product version information used for an FDI Communication Server.

J.6 Associating a Package with an FDI Communication Server

An OPC UA based FDI Communication Server is uniquely identified by its ProductUri. The mapping of the catalog information shall be in accordance with Table J.1.

Table J.1 – Catalog Mapping

Catalog Element	OPC UA Mapping
ProductUri	ProductUri

J.7 Handling of Catalog elements

Some parts of the catalog need to be handled in accordance with Table J.2.

Table J.2 – Handling of Catalog elements

Catalog Element	Handling
ClassificationId	"NETWORK"
ListOfSupportedDeviceRevisions	XML Element not provided

J.8 Example

An example for /fdicatalog/catalog.xml of an FDI Communication Server is listed below.

```
<?xml version="1.0" encoding="UTF-8"?>
<fdi:Catalog
xmlns:fdi="http://fdi-cooperation.com/2010/package"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://fdi-cooperation.com/2010/package
catalog.xsd">
  <PackageId>f516f651-3e0f-4672-bcfe-67a4141a7a25</PackageId>
  <PackageType>Communication</PackageType>
  <Version>1.0.0</Version>
  <FdiVersionSupported>1.0.0</FdiVersionSupported>
  <ManufacturerName>Communication Provider GmbH</ManufacturerName>
  <ManufacturerContact>Hauptstrasse 17, Neustadt,
Germany</ManufacturerContact>
  <ManufacturerUrl>http://cpg.local</ManufacturerUrl>
  <ManufacturerImage>rIdMfrLogo</ManufacturerImage>
  <CommunicationServer>
    <ProductUri>urn:cpg:comserver</ProductUri>
  </CommunicationServer>
  <ListOfDeviceTypes>
    <DeviceType>
      <Name>
        <value>FDI Communication Server for HART</value>
        <value xml:lang="de">FDI Kommunikationsserver für HART</value>
      </Name>
      <ClassificationId>NETWORK</ClassificationId>
      <ListOfInterfaces>
        <Interface>
          <ListOfCommunicationProfiles>
            <CommunicationProfile>hart_fsk</CommunicationProfile>
          </ListOfCommunicationProfiles>
          <Version>5.0.0</Version>
          <CommunicationRole>SERVER</CommunicationRole>
        </Interface>
      </ListOfInterfaces>
      <Edd>rIDEDD</Edd>
      <ListOfImages>
        <Image>rIdPicture1</Image>
        <Image>rIdPicture2</Image>
      </ListOfImages>
      <ListOfDocuments>
        <Document>rIdDocument1</Document>
      </ListOfDocuments>
    </DeviceType>
  </ListOfDeviceTypes>
</fdi:Catalog>
```

Annex K
(normative)

FDI Profile for EDDs

K.1 Overview

Annex K describes rules that need to be applied to an EDD in order to fulfil the conformance to the FDI profile for EDDs. Annex K does not define new EDD concepts or constructs, but only defines that some optional constructs defined in the EDD specification are mandatory and some other concepts shall not be used in order to be compliant with the FDI profile for EDDs.

K.2 Entry Point to Online handling

The EDD shall contain at least one entry point for online handling (device_root_menu, diagnostic_root_menu, maintenance_root_menu or process_variables_root_menu).

K.3 Entry Point to Offline handling

The EDD shall contain at least one entry point for offline handling by providing the offline_root_menu.

K.4 Upload and Download

The EDD shall contain an upload menu (upload_from_device_root_menu or download_variables). The EDD shall contain a download menu (download_to_device_root_menu). The upload and download menus shall not contain any user interactions (call to User Interface Builtins).

K.5 Initial Data Set

The EDD shall provide a valid initial data set for offline configuration without being connected to the device. There shall be at least one device variant where this configuration could be directly downloaded without modifications.

This can be achieved by using mechanisms defined in EDDL (e.g. INITIAL_VALUE, DEFAULT_VALUE) or by using the defaults of the respective data types.

NOTE EDD offers additional concepts to create valid offline configurations, such as TEMPLATEs. Those can be used to create different variants of initial settings.

K.6 Method GetHealthStatus

The EDD shall include the GetHealthStatus method to provide access to health status state. See Annex H.

K.7 Actions

K.7.1 Pre- and Post-Read Actions

The pre- and post-read actions (PRE_READ_ACTIONS and POST_READ_ACTIONS) on VARIABLEs or MENUs shall not contain any user interactions (call to User Interface Builtins).

K.7.2 Pre- and post-write actions

The pre- and post-write actions (PRE_WRITE_ACTIONS and POST_WRITE_ACTIONS) on VARIABLEs or MENUs shall not contain any user interactions (call to User Interface Builtins).

K.7.3 Refresh actions on variables

The refresh actions (REFRESH_ACTIONS) on VARIABLEs shall not contain any user interactions (call to User Interface Builtins).

NOTE Other refresh actions (e.g. on graphs) can have calls to User Interface Builtins.

K.7.4 Actions on BIT_ENUMERATION

Actions on BIT_ENUMERATION shall not contain any user interactions (call to User Interface Builtins).

K.8 Shared files

Use of shared files (using SHARED on the FILE construct) is not recommended and will be ignored in FDI Hosts.

NOTE Future versions of the FDI Technology may support this feature.

Bibliography

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 62591, *Industrial communication networks – Wireless communication network and communication profiles – WirelessHART*

IEC 62769-2, *FDI Project Technical Specification – Part 2: FDI Client*

IEC 62769-3, *FDI Project Technical Specification – Part 3: FDI Server*

IEC 62769-5, *FDI Project Technical Specification – Part 5: FDI Information Model*

IEC 62769-7, *FDI Project Technical Specification – Part 7: FDI Communication Devices*

NE107, *Self-Monitoring and Diagnosis of Field Devices,* available at www.namur.de

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 4: Paquetages FDI

AVANT-PROPOS

1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62769-4 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Cette deuxième édition annule et remplace la première édition parue en 2015. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

a) prise en charge des Développeurs de Paquetages pour la construction d'EDD ciblés pour le système de base actuel au moyen d'un outil unique de développement;
b) la signature numérique comprend désormais un horodatage de confiance pour la validation sur le long terme du paquetage FDI;
c) horodatage pour la signature de paquetage d'appareils.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
65E/761/FDIS	65E/771/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

La version française de la norme n'a pas été soumise au vote.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62769, publiées sous le titre général *Intégration des appareils de terrain (FDI)*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "http://webstore.iec.ch" dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La série IEC 62769 est publiée sous le titre général "*Intégration des appareils de terrain* (*FDI*)" et comporte les parties suivantes:

– Partie 1: Vue d'ensemble
– Partie 2: Client FDI
– Partie 3: Serveur FDI
– Partie 4: Paquetages FDI
– Partie 5: Modèle d'Information FDI
– Partie 6: Mapping de technologies FDI
– Partie 7: Appareils de Communication FDI
– Partie 100: Profils – Extensions de protocoles génériques
– Partie 101-1: Profils – Foundation Fieldbus H1
– Partie 101-2: Profils – Foundation Fieldbus HSE
– Partie 103-1: Profils – PROFIBUS
– Partie 103-4: Profils – PROFINET
– Partie 109-1: Profils – HART et WirelessHART
– Partie 115-2: Profils – Définitions spécifiques au protocole pour Modbus-RTU
– Partie 150-1: Profils – ISA 100.11a

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 4: Paquetages FDI

1 Domaine d'application

La présente partie de l'IEC 62769 spécifie les Paquetages FDI. L'architecture FDI complète est représentée à la Figure 1. Les composants architecturaux qui relèvent du domaine d'application du présent document ont été mis en évidence dans la Figure 1.

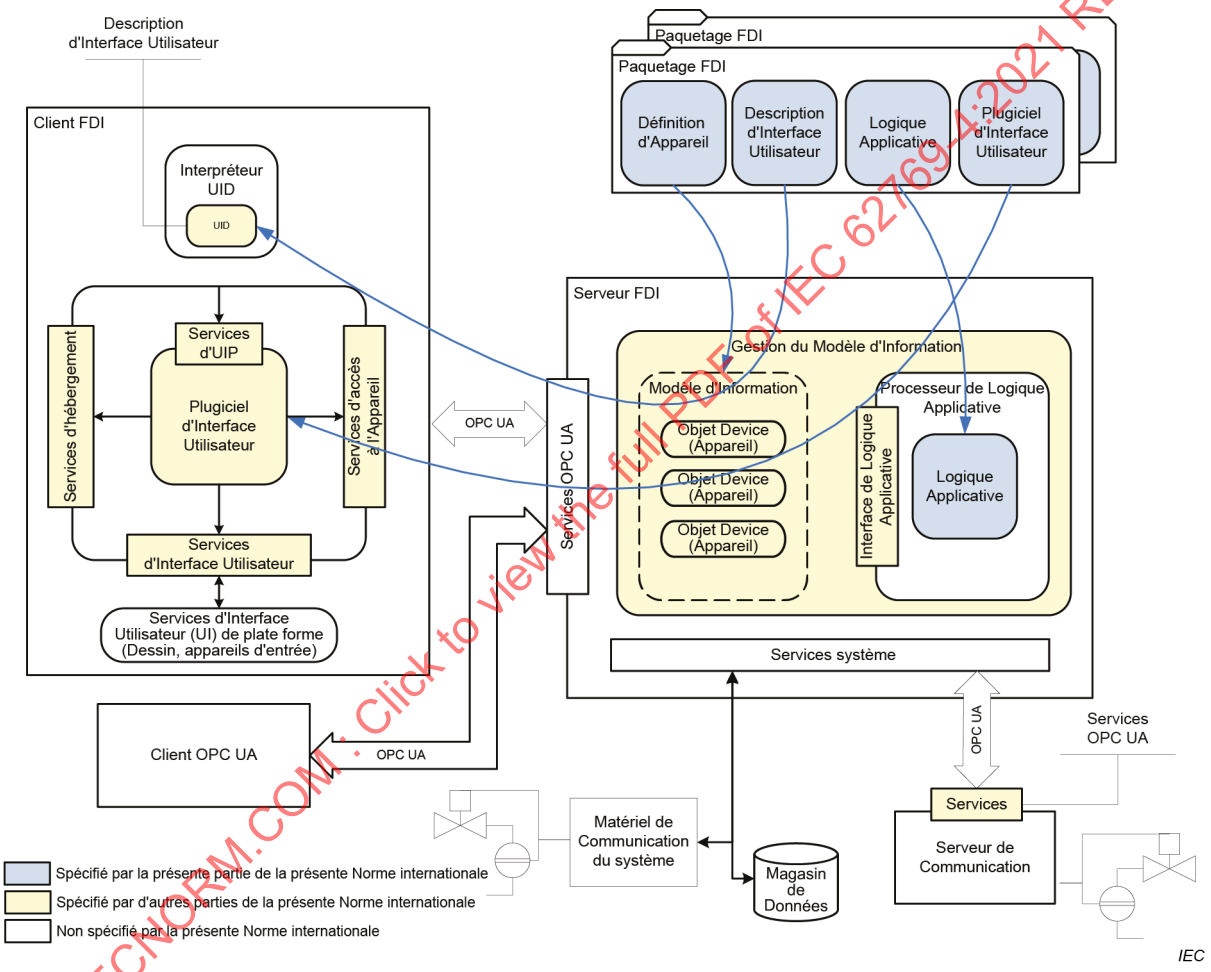

Figure 1 – Diagramme de l'architecture FDI

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61804 (toutes les parties), *Blocs fonctionnels (FB) pour les procédés industriels et le langage de description électronique de produit (EDDL)*

IEC 61804-5:2015, *Blocs fonctionnels (FB) pour les procédés industriels et le langage de description électronique de produit (EDDL) – Partie 5: Bibliothèque de Builtin EDDL*

IEC 62769-1, *Intégration des appareils de terrain (FDI) – Partie 1: Vue d'ensemble*

IEC 62769-6, *Intégration des appareils de terrain (FDI) – Partie 6: Mapping de technologies FDI*

ISO/IEC 29500-2:2016, *Information technology – Document description and processing languages – Office Open XML File Formats – Part 2: Open Packaging Conventions* (disponible en anglais seulement)

ISO 639-1, *Codes pour la représentation des noms de langue – Partie 1: Code alpha-2*

ISO 32000-1, *Document management – Portable document format – Part 1: PDF 1.7* (disponible en anglais seulement)

Extensible Markup Language (XML) 1.0, W3C Recommendation, disponible à l'adresse <http://www.w3.org/TR/REC-xml/>

XML Schema Definition Language (XSD) 1.1, W3C Recommendation, disponible à l'adresse <http://www.w3.org/TR/xmlschema11-1/>

ETSI EN 319 132-1, *Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures* (disponible en anglais seulement)

ETSI TS 101 733, *Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)* (disponible en anglais seulement)

FIPS 140-2, Security Requirements for Cryptographic Modules

3 Termes, définitions, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 62769-1 et de l'ISO/IEC 29500-2, ainsi que les suivants, s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

3.1.1
Pièce Jointe
fichiers de prise en charge spécifiques à l'appareil et au protocole qui ne sont pas directement utilisés pour intégrer l'Appareil dans le système

3.1.2
paquetage de communication FDI
paquetage FDI qui fournit des informations destinées à intégrer un Serveur de Communication FDI à un Serveur FDI

Note 1 à l'article: L'abréviation "FDI" est dérivée du terme anglais développé correspondant *"Field Device Integration"*.

3.1.3
paquetage d'appareil FDI
paquetage FDI qui fournit un ou plusieurs types d'appareils à un Serveur FDI

3.1.4
modèle de paquetage FDI
description de la structure et des éléments d'un Paquetage FDI

3.1.5
paquetage de profil FDI
paquetage FDI qui fournit une information pour créer un nœud de type d'appareil qui peut être associé à une classe d'appareils

3.1.6
autorité d'enregistrement de FDI
entité qui a le droit et la capacité d'effectuer des essais de conformité FDI sur des Paquetages FDI et de délivrer des documents de certificat d'enregistrement

3.1.7
paquetage d'UIP FDI
paquetage FDI qui fournit un ou plusieurs UIP à un Serveur FDI

Note 1 à l'article: L'abréviation "UIP" est dérivée du terme anglais développé correspondant *"User Interface Plug-in"*.

3.1.8
catalogue de paquetage
fichier qui décrit le contenu d'un Paquetage FDI

3.1.9
catalogue d'UIP
fichier qui décrit les propriétés d'un UIP

3.1.10
variante d'UIP
élément spécifique à une plate-forme d'un Plugiciel d'Interface Utilisateur

Note 1 à l'article: Un UIP est composé d'une ou de plusieurs variantes. Par exemple, une variante peut être optimisée pour des appareils portables, tandis qu'une autre variante est optimisée pour des appareils à grand écran.

3.2 Termes abrégés

Pour les besoins du présent document, les abréviations de l'IEC 62769-1, ainsi que les suivantes s'appliquent.

CFF	Capabilities File for FOUNDATION Fieldbus (Fichier Capacités pour bus de terrain de la FOUNDATION Fieldbus)
ID	Identificateur
IDE	Integrated Development Environment (Environnement de développement intégré)
IM	Information Model (Modèle d'Information)
PNG	Portable Network Graphics (Format graphique de réseaux portables)
ZIP	Zipper (format de fichier permettant l'archivage)

3.3 Conventions

Pour les besoins du présent document, les conventions de l'IEC 62769-1 s'appliquent.

4 Modèle de Paquetage FDI

4.1 Vue d'ensemble

Le modèle de Paquetage FDI (voir la Figure 2) fournit tous les éléments nécessaires pour intégrer les appareils, les composants réseau et les Serveurs de Communication FDI dans un système.

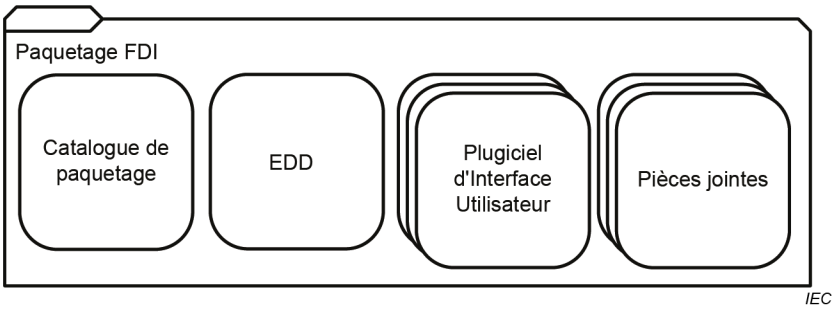

Figure 2 – Modèle de Paquetage FDI

La Figure 3 représente le mapping des éléments fonctionnels du Paquetage FDI, tel que spécifié dans l'IEC 62769-1, aux éléments physiques contenus dans un Paquetage FDI réel, comme spécifié dans le présent document.

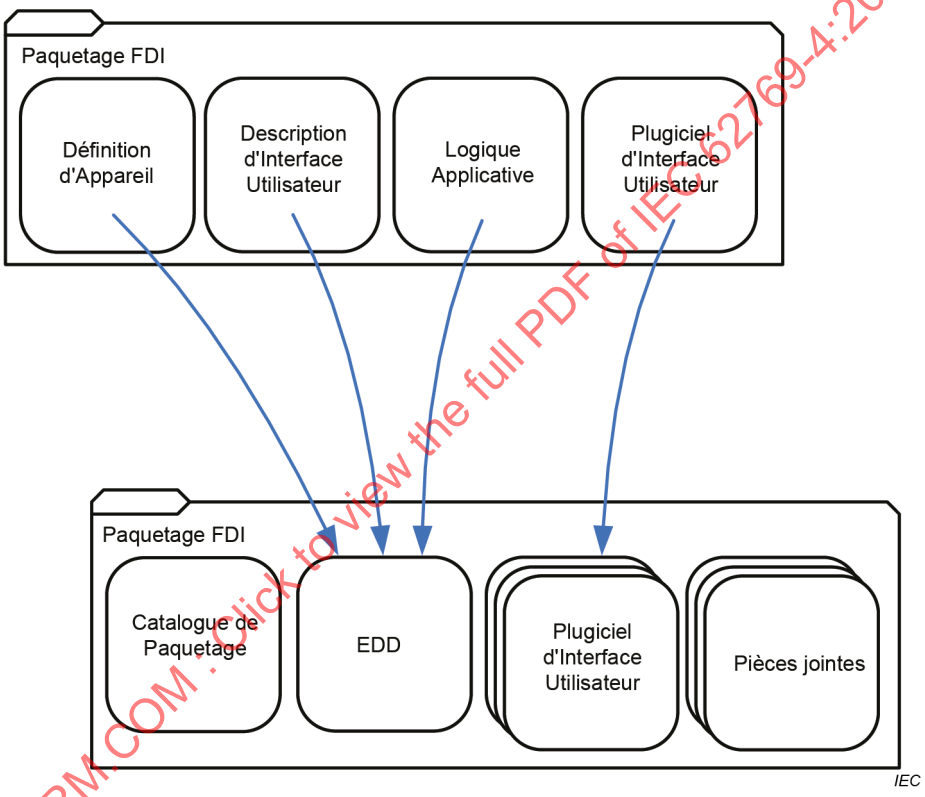

Figure 3 – Mapping architectural

La Description d'Appareil Électronique (EDD – *Electronic Device Description*) correspond aux éléments fonctionnels de la Définition d'Appareil, de la Description d'Interface Utilisateur et de la Logique Applicative. Un ensemble de Plugiciels d'Interface Utilisateur physiques correspond à l'élément fonctionnel d'un Plugiciel d'Interface Utilisateur.

Les autres éléments physiques contenus dans le Paquetage FDI, le Catalogue de Paquetage et les Pièces jointes, assurent une prise en charge du fabricant pour les mécanismes de base tels que l'identification, la gestion de version, le déploiement et la distribution et/ou des informations spécifiques au protocole relatives à l'appareil et/ou aux Plugiciels d'Interface Utilisateur.

4.2 Éléments de Paquetages FDI

4.2.1 Catalogue de Paquetage

Le Catalogue de Paquetage est un élément exigé qui fournit des informations sur le contenu du Paquetage FDI, y compris, entre autres, des informations sur l'identification, les versions et le type d'appareil, les exigences du matériel et de l'appareil E/S, la Version de Technologie FDI et les caractéristiques spécifiques au protocole.

Le Catalogue de Paquetage est utilisé par un Serveur FDI pour créer un catalogue des types d'appareils et des Nœuds de DeviceType dans le Modèle d'Information.

4.2.2 Description d'Appareil Électronique (EDD)

La Description d'Appareil Électronique (EDD) est un élément qui fournit la Définition d'Appareil, les Descriptions d'Interface Utilisateur et la Logique Applicative à un Serveur FDI.

Le paragraphe 4.3 spécifie les types de Paquetages FDI pour lesquels une EDD est exigée.

Pour optimiser l'interopérabilité, la configuration initiale d'un appareil doit être réalisable uniquement avec les Descriptions d'Interface Utilisateur, la Définition d'Appareil et la Logique Applicative qui font partie de l'EDD dans le Paquetage d'Appareil FDI. L'utilisation des Plugiciels d'Interface Utilisateur est facultative et est ciblée en particulier pour la mise en place d'appareils complexes.

4.2.3 Plugiciel d'Interface Utilisateur (UIP)

Un Plugiciel d'Interface Utilisateur (UIP) est un élément qui permet à un Client FDI de présenter une interface utilisateur programmée. Le Serveur FDI stocke uniquement l'UIP issu d'un paquetage consommé; il n'exécute pas ni n'interprète l'UIP.

Un UIP est référencé à partir d'une Description d'Interface Utilisateur. Ces références se reflètent dans le Modèle d'Information à travers des groupes fonctionnels (voir l'IEC 62769-5). Les références aux UIP dans le Modèle d'Information sont logiques à travers une référence unique. La structure physique de l'UIP n'est pas au niveau du Modèle d'Information. Physiquement, un seul UIP peut être constitué d'une ou de plusieurs variantes d'UIP, chacune ciblant une plate-forme et un environnement d'exécution spécifiques.

La Figure 4 représente la référence logique à un UIP dans l'EDD et la structure physique de cet UIP. L'EDD du Paquetage FDI référence un identificateur unique au niveau global de l'UIP. En outre, le Catalogue de Paquetage énumère également les UIP exigés par le type d'appareil ainsi que les versions d'UIP prises en charge par le paquetage.

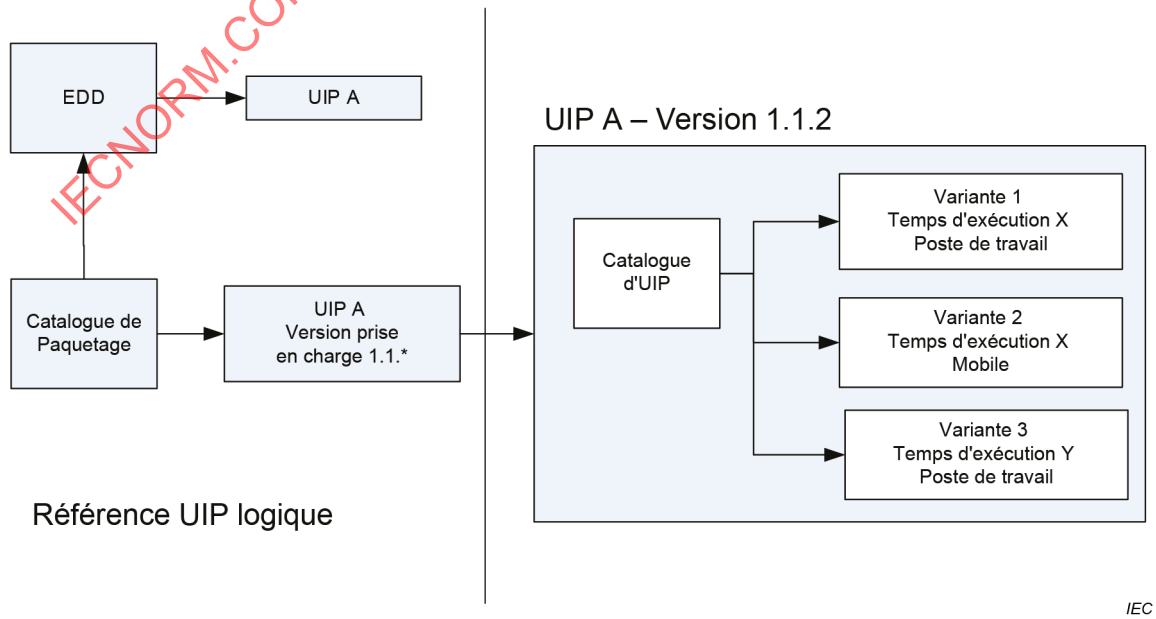

Figure 4 – Modèle de Référence du Plugiciel d'Interface Utilisateur

Un UIP peut être conçu pour satisfaire aux différentes exigences de la plate-forme étant donné que toutes les plates-formes ne prennent pas en charge les mêmes tailles d'écran et les appareils d'entrée.

Lorsqu'un Client FDI demande à un UIP de spécifier le type de plate-forme, le mécanisme de déploiement au sein du Serveur FDI cherche une Variante d'UIP qui correspond au type de plate-forme spécifié et la renvoie au Client FDI.

La plate-forme cible définit les résolutions d'écran spécifiques et les appareils d'entrée qui doivent être pris en charge par la Variante d'UIP. Les plates-formes cibles disponibles sont décrites dans le Tableau 1.

Tableau 1 – Plate-forme d'UIP

Plate-forme	Caractéristiques
Poste de travail	Client complet avec, en général, un grand écran, un clavier complet et une entrée de souris.
Mobile	Client limité avec, en général, un petit écran et des capacités d'entrée limitées.

Les exigences détaillées de la plate-forme dépendent de la technologie et sont spécifiées dans l'IEC 62769-6.

Un UIP peut être livré en interne (autonome) ou l'UIP peut être livré au moyen d'un Paquetage d'UIP FDI. Une EDD d'un paquetage ne doit pas faire référence à un UIP livré autonome avec un autre paquetage.

La Figure 5 représente deux Paquetages d'Appareil FDI qui référencent des UIP à partir d'un Paquetage d'UIP FDI. Dans cet exemple, l'EDD du Paquetage d'Appareil A et du Paquetage d'Appareil B référence l'UIP B livré par le Paquetage d'UIP C.

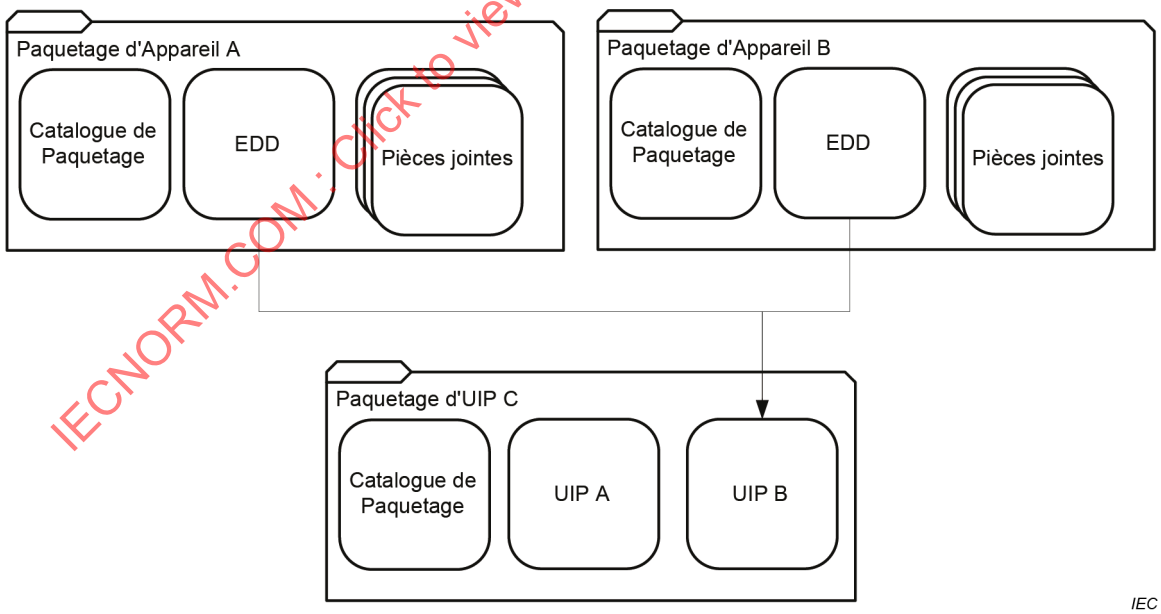

Figure 5 – Multiples Paquetages FDI référençant un UIP commun

4.2.4 Pièce jointe

Les pièces jointes fournissent un appareil et des fichiers de prise en charge spécifiques au protocole, ainsi que d'autres fichiers qui ne sont pas directement utilisés pour intégrer l'appareil dans le système, par exemple, des manuels de produits.

Les types suivants de Pièces jointes sont définis et pris en compte dans le Catalogue de Paquetage:

- Fichiers spécifiques au protocole (voir 5.3.4.3);
- Documentation (voir 5.3.4.2);
- Images de l'appareil (voir 5.3.4.1).

Certains fichiers spécifiques au protocole sont obligatoires (voir l'Annex F).

4.3 Types de Paquetages FDI

4.3.1 Paquetage d'Appareil FDI

Le Paquetage d'Appareil FDI est destiné à fournir à un système des informations relatives à un appareil. Un Paquetage d'Appareil FDI décrit un seul type d'appareil. La

Figure 6 représente la structure physique d'un Paquetage d'Appareil FDI qui décrit un seul type d'appareil. Des détails sur la façon de créer un paquetage pour un appareil modulaire sont donnés dans l'Annex I.

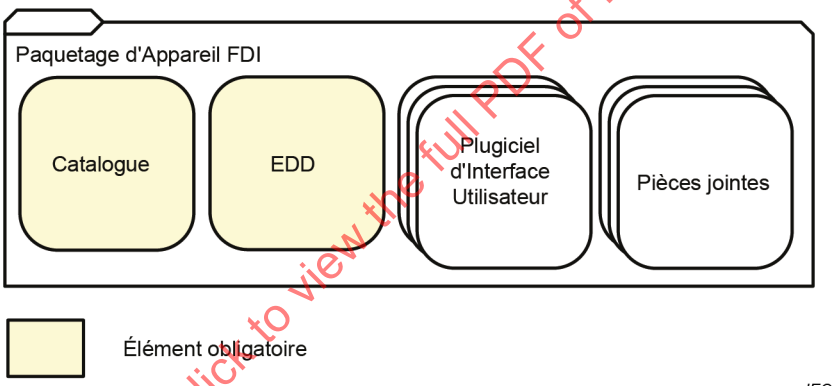

Figure 6 – Paquetage d'Appareil FDI

Le Paquetage d'Appareil FDI doit avoir un seul Catalogue de Paquetage.

Un Paquetage FDI pour des Appareils simples contient une EDD. Pour les Appareils modulaires, plusieurs EDD peuvent exister (voir l'Annex I).

Le Paquetage d'Appareil FDI peut inclure un ou plusieurs Plugiciels d'Interface Utilisateur.

Le Paquetage d'Appareil FDI peut inclure des Pièces jointes.

4.3.2 Paquetage de Communication FDI

Le Paquetage de Communication FDI est destiné à fournir des informations relatives à un seul Appareil de Communication FDI. L'ensemble des Appareils de Communication FDI se divise en deux sous-groupes.

– Paquetages de Communication FDI pour les Passerelles – Contiennent tous les éléments exigés pour la description de toutes les fonctions d'appareils, ainsi que la logique exigée pour le pontage des réseaux utilisant différents protocoles de communication, notamment les algorithmes de pontage (des informations détaillées sont décrites dans l'IEC 62769-7).

– Paquetages de Communication FDI pour les Serveurs de Communication PDI – Contiennent les informations fondamentales pour décrire l'appareil de communication et l'intégrer à l'Hôte FDI, mais également pour créer une référence valide vers l'application du Serveur de Communication FDI externe. Cette application ne doit pas être fournie dans le cadre du Paquetage de Communication FDI. Le Serveur de Communication FDI fournit un accès aux appareils de terrain individuels ou aux réseaux d'appareils de terrain (des informations détaillées sont décrites dans l'IEC 62769-7).

  Les exigences relatives au contenu du Paquetage de Communication FDI spécifiques aux Serveurs de Communication FDI sont définies à l'Annex J.

Toutefois, les représentants des deux groupes sont intégrés aux Hôtes FDI en utilisant des Paquetages de Communication FDI. Leur structure physique est représentée à la Figure 7. Les composants binaires nécessaires pour communiquer avec le matériel de communication doivent être fournis en dehors du domaine d'application de ce Paquetage de Communication FDI.

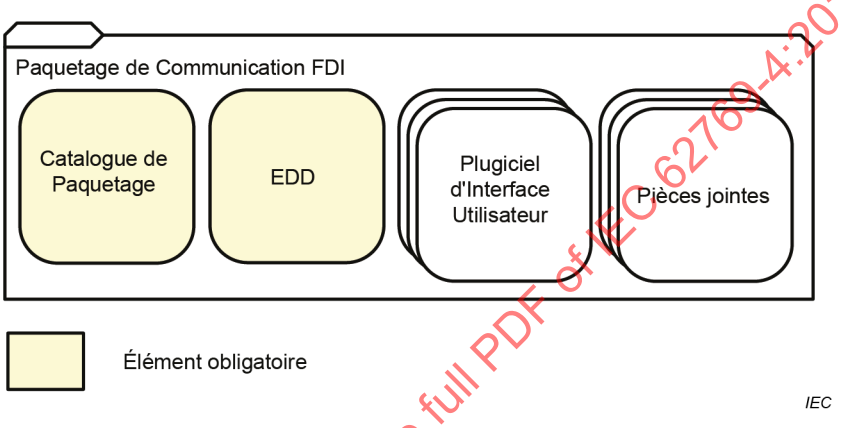

Figure 7 – Paquetage de Communication FDI

4.3.3 Paquetage d'UIP FDI

Le Paquetage d'UIP FDI délivre des Plugiciels d'Interface Utilisateur à un Serveur FDI. Il est utilisé pour distribuer un ou plusieurs UIP qui sont destinés à être partagés par plusieurs types d'appareils. Sa structure physique est représentée à la Figure 8.

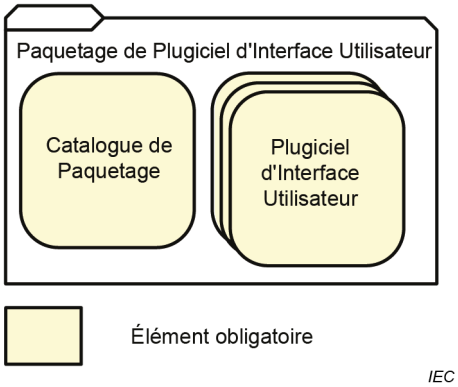

Figure 8 – Paquetage d'UIP FDI

Le Paquetage d'UIP FDI doit avoir un Catalogue de Paquetage et un ou plusieurs Plugiciels d'Interface Utilisateur.

4.3.4 Paquetage de profil FDI

Un Paquetage de Profil FDI fournit des informations pour la création d'un nœud de type d'appareil qui peut être associé à une classe d'appareils mettant en œuvre un ensemble commun de paramètres et de fonctions (décrit dans le ou les profils comme des définitions créées par exemple par des fondations de communication ou des groupes d'intérêts similaires).

Conceptuellement, un Paquetage de Profil FDI fournit une information qui est une hyperclasse d'un Paquetage d'Appareil FDI. Le Paquetage de Profil FDI est représenté à la Figure 9.

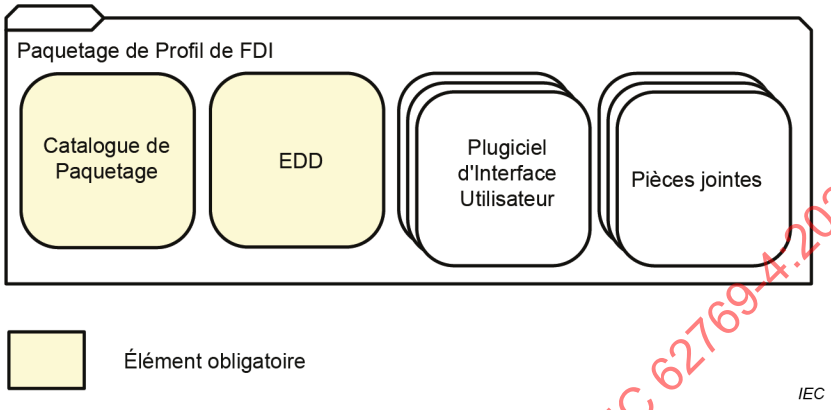

Figure 9 – Paquetage de Profil FDI

Ces paquetages permettent l'intégration des appareils sans avoir un Paquetage d'Appareil FDI spécifique. Seuls les Paquetages de Profil FDI prennent en charge les fonctionnalités normalisées telles que définies par le protocole de communication et les profils d'application correspondants (Annex F).

Un Paquetage de Profil FDI doit être autonome. Par exemple, si un Profil B élargit la définition d'un Profil A, le Paquetage de Profil FDI pour le Profil B doit alors inclure une définition complète du Profil B, y compris les éléments provenant du Profil A.

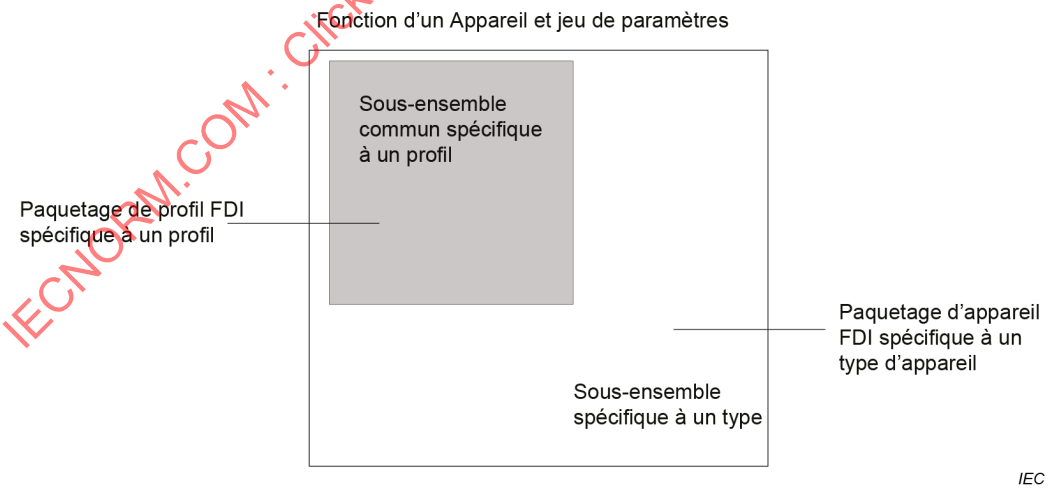

Figure 10 – Fonction d'un Appareil et jeux de paramètres (spécifiques au type et au profil)

La Figure 10 représente l'ensemble des fonctionnalités d'un appareil. Certaines des fonctionnalités se conforment à un profil de communication spécifique, qui peut être accessible à l'aide d'un Paquetage de Profil FDI. S'il est nécessaire d'accéder à une fonctionnalité spécifique à un Type d'Appareil, en plus de la fonctionnalité commune, alors un Paquetage d'Appareil FDI est exigé.

La description détaillée des exigences spécifiques au protocole de communication ne fait pas partie du présent document.

Généralement, une mise en œuvre de la sélection automatique d'un Paquetage FDI (lancement de Paquetages FDI correspondant à un appareil physique connecté) cherche tout d'abord des Paquetages d'Appareil FDI spécifiques à un Type d'Appareil disponible. Si le paquetage spécifique n'est pas disponible, le Paquetage de Profil FDI spécifique à un profil est chargé (si disponible). Ce comportement est cependant spécifique à un système.

5 Mise en œuvre d'un Paquetage FDI

5.1 Technologie de Paquetage

Le format du Paquetage FDI est conforme à la Convention de Paquetage Ouvert (Open Packaging Convention) spécifiée par l'ISO/IEC 29500-2. La spécification de la Convention de Paquetage Ouvert est conçue pour représenter de très nombreuses applications. La technologie a été conçue pour mettre en place un mécanisme assigné à la fourniture, au déploiement et à l'utilisation d'un ensemble de composants logiquement et physiquement reliés d'une manière souple, extensible, efficace et ouverte.

Le paragraphe 5.2 clarifie l'utilisation des éléments spécifiques de la spécification des Conventions de Paquetage Ouvert.

Les conventions de dénomination applicables au fichier de paquetage FDI sont décrites à l'Annex A. Les principes fondamentaux de création d'un paquetage FDI sont décrits à l'Annex B. Un exemple de mise en œuvre de paquetage de dispositif FDI est décrit à l'Annex D.

NOTE L'adresse *fdi-cooperation.com* a été utilisée lorsque la FDI Cooperation LLC existait. L'adresse universelle n'a pas été modifiée afin de maintenir une certaine cohérence. L'adresse universelle n'indique pas de propriété intellectuelle technologique et peut ne pas conduire à une adresse valide.

5.2 Utilisation des Conventions de Paquetage Ouvert

5.2.1 Parties inconnues

Les parties inconnues sont toutes les parties qui ne sont pas définies dans le présent document ou dans l'ISO/IEC 29500-2. Les parties inconnues peuvent exister, car la spécification FDI est mise à jour et le Serveur FDI hérité peut ne pas avoir connaissance de ces parties supplémentaires. Les parties inconnues doivent être ignorées par le Serveur FDI pour permettre la compatibilité ascendante.

5.2.2 Parties invalides

Les parties invalides sont les parties qui ne sont pas conformes aux lignes directrices de dénomination ou qui ne sont pas associées aux types de contenu spécifiés dans le présent document ou dans l'ISO/IEC 29500-2. Les parties invalides ne doivent pas être utilisées dans un Paquetage FDI.

5.2.3 Relations inconnues

Les relations inconnues sont celles qui ne sont pas définies dans le présent document ou dans l'ISO/IEC 29500-2. Les relations inconnues doivent être ignorées par un consommateur du Paquetage FDI.

5.2.4 Entrelacement

Toutes les parties du Paquetage FDI doivent être établies à l'aide d'un ordonnancement simple. Le producteur de paquetage ne doit pas modifier l'ordre des parties.

5.2.5 Propriétés principales

Les propriétés principales ne sont pas utilisées dans le Modèle de Paquetage FDI et doivent être ignorées par tous les consommateurs de Paquetage FDI.

5.2.6 Miniatures

La miniature (thumbnail) est une partie facultative d'un Paquetage FDI.

5.2.7 Signatures numériques

L'utilisation de signatures numériques telles que spécifiées dans l'ISO/IEC 29500-2 est une partie obligatoire d'un Paquetage FDI. Pour de plus amples informations, voir l'Article 7.

5.3 Parties de Paquetage FDI

5.3.1 Catalogue de Paquetage

5.3.1.1 Format

La partie Catalogue de Paquetage est un fichier XML dont le schéma est défini à l'Annex E. Un Paquetage FDI doit avoir un seul Catalogue de Paquetage. Le Catalogue de Paquetage doit être identifié par une seule relation de paquetage. L'élément racine du fichier est Catalogue. La structure d'un élément Catalogue est représentée à la Figure 11.

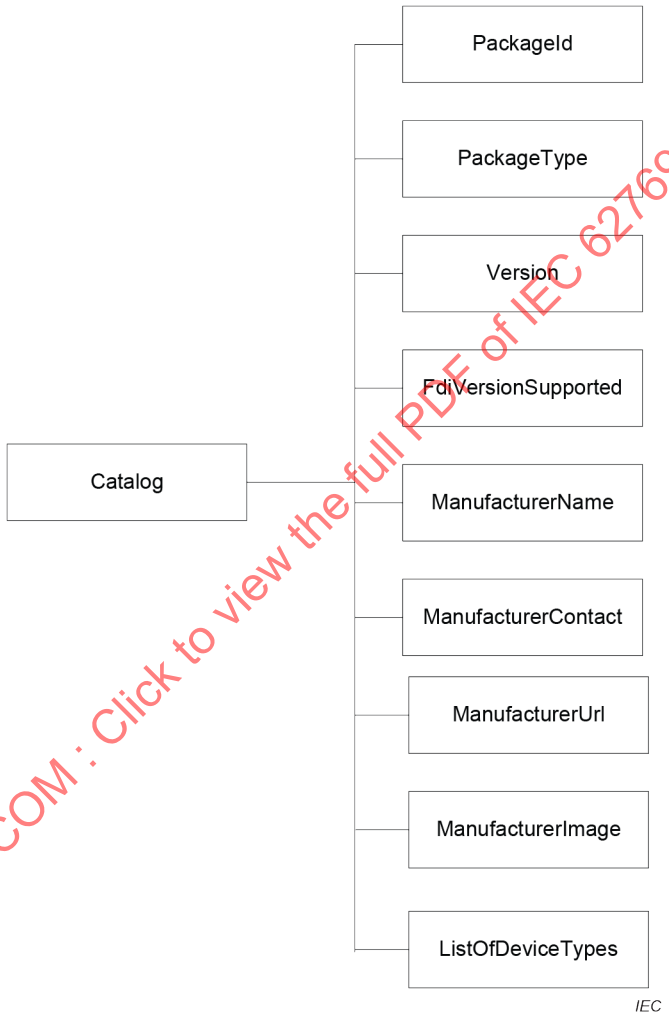

Figure 11 – Élément du Catalogue

La partie Catalogue de Paquetage est décrite dans le Tableau 2.

Tableau 2 – Partie Catalogue de Paquetage

Partie	Contenu
Type de contenu	application/vnd.fdi.package.catalog+xml
Espace de noms racine	http://fdi-cooperation.com/2010/package-catalog
Relation source	http://fdi-cooperation.com/2010/relationships/package-catalog
Nom de fichier	catalog.xml

5.3.1.2 Contenu

Le schéma pour le Catalogue de Paquetage est décrit à l'Annex E.

5.3.2 Description d'Appareil Électronique (EDD)

5.3.2.1 Format

L'EDD est un fichier codé qui utilise le Langage de Description d'Appareil Électronique (EDDL) selon 5.3.2.2.

Le format de la partie EDD est décrit dans le Tableau 3.

Tableau 3 – Partie EDD

Partie	Contenu
Type de contenu	application/vnd.fdi.package.edd
Espace de noms racine	Non applicable
Relation source	http://fdi-cooperation.com/2010/relationships/edd
Nom de fichier	Non spécifié

5.3.2.2 Contenu

L'EDDL est spécifié dans l'IEC 61804. Les règles spécifiques à la FDI et applicables aux EDD sont décrites à l'Annex K.

5.3.3 Plugiciel d'Interface Utilisateur (UIP)

5.3.3.1 Format

Un Paquetage FDI peut contenir une ou plusieurs parties Plugiciel d'Interface Utilisateur. Toutes les parties Plugiciel d'Interface Utilisateur doivent être identifiées par une relation de paquetage.

Le Plugiciel d'Interface Utilisateur représente un élément conteneur pour les Variantes d'UIP (voir 5.3.3.2.2.2), qui contient la représentation physique d'un UIP (des variantes différentes pour des plates-formes différentes) qui est consommé et exécuté par un Client FDI. Un Plugiciel d'Interface Utilisateur doit au moins fournir une Variante d'UIP.

Les variantes sont empaquetées dans un seul Plugiciel d'Interface Utilisateur utilisant une Convention de Paquetage Ouvert comme spécifié en 5.3.3.2. Le Plugiciel d'Interface Utilisateur n'est pas directement consommé par un Serveur FDI. Le Plugiciel d'Interface Utilisateur est une partie facultative de différents types de Paquetages FDI comme spécifié en 4.3. Le Plugiciel d'Interface Utilisateur est représenté à la Figure 12.

Les informations relatives à la version d'UIP ainsi que le comportement de mise à jour et de mise à niveau doivent suivre les règles et concepts décrits dans l'IEC 61804-1.

Le paramètre régional par défaut pour les UIP et toutes les variantes contenues est: Anglais (US). La prise en charge de langues facultatives est admise selon les besoins du marché.

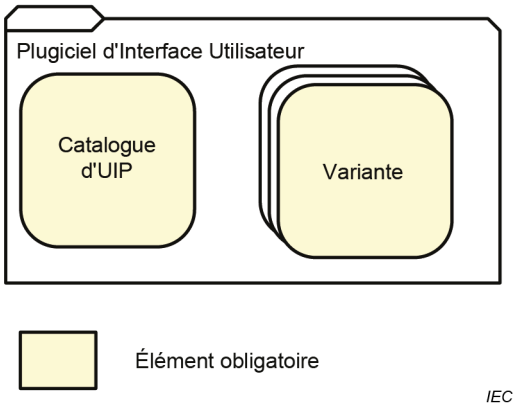

Figure 12 – Plugiciel d'Interface Utilisateur

Le format de la partie Plugiciel d'Interface Utilisateur est décrit dans le Tableau 4.

Tableau 4 – Partie Plugiciel d'Interface Utilisateur

Partie	Contenu
Type de contenu	application/vnd.fdi.package.uip
Espace de noms racine	Non applicable
Relation source	http://fdi-cooperation.com/2010/relationships/uip
Nom de fichier	l'extension doit être .uip

5.3.3.2 Contenu

5.3.3.2.1 Utilisation d'une Convention de Paquetage Ouvert

5.3.3.2.1.1 Propriétés principales

Les propriétés principales ne sont pas utilisées et doivent être ignorées par tous les consommateurs de Paquetage d'UIP FDI.

5.3.3.2.1.2 Miniatures

Les miniatures ne sont pas utilisées et doivent être ignorées par tous les consommateurs de Paquetage d'UIP FDI.

5.3.3.2.1.3 Signatures numériques

Les signatures numériques peuvent être nécessaires pour les Variantes d'UIP (voir 5.3.3.2.2.2) en fonction de la technologie de mise en œuvre. Les informations détaillées de la technologie de mise en œuvre et les mécanismes nécessaires à mettre en œuvre sont décrits dans l'IEC 62769-6.

Le Plugiciel d'Interface Utilisateur en tant que tel ne doit toutefois pas être signé ou alors les signatures appliquées doivent être ignorées par le composant utilisateur de FDI.

5.3.3.2.2 Parties de Plugiciel d'Interface Utilisateur

5.3.3.2.2.1 Catalogue d'UIP

####### 5.3.3.2.2.1.1 Format

Le Catalogue d'UIP est un fichier XML qui décrit les propriétés d'un UIP qui sont nécessaires pour créer les nœuds du Modèle d'Information dans un côté serveur FDI et pour le déploiement (Serveur FDI à Client FDI).

Le Catalogue d'UIP héberge également des informations relatives à l'élément physique de démarrage qui doit être appelé/exécuté à partir d'une Variante d'UIP sur le Client FDI. Le format et le type de cet élément de démarrage sont spécifiques à une technologie et donc définis dans l'IEC 62769-6.

Le Catalogue d'UIP doit être identifié par une seule relation de paquetage.

La partie de Catalogue d'UIP du paquetage est un fichier XML dont le schéma est défini dans l'Annex E. Un Plugiciel d'Interface Utilisateur doit avoir un seul Catalogue d'UIP. L'élément racine du fichier est UipCatalog. La structure de l'élément UipCatalog est représentée à la Figure 13.

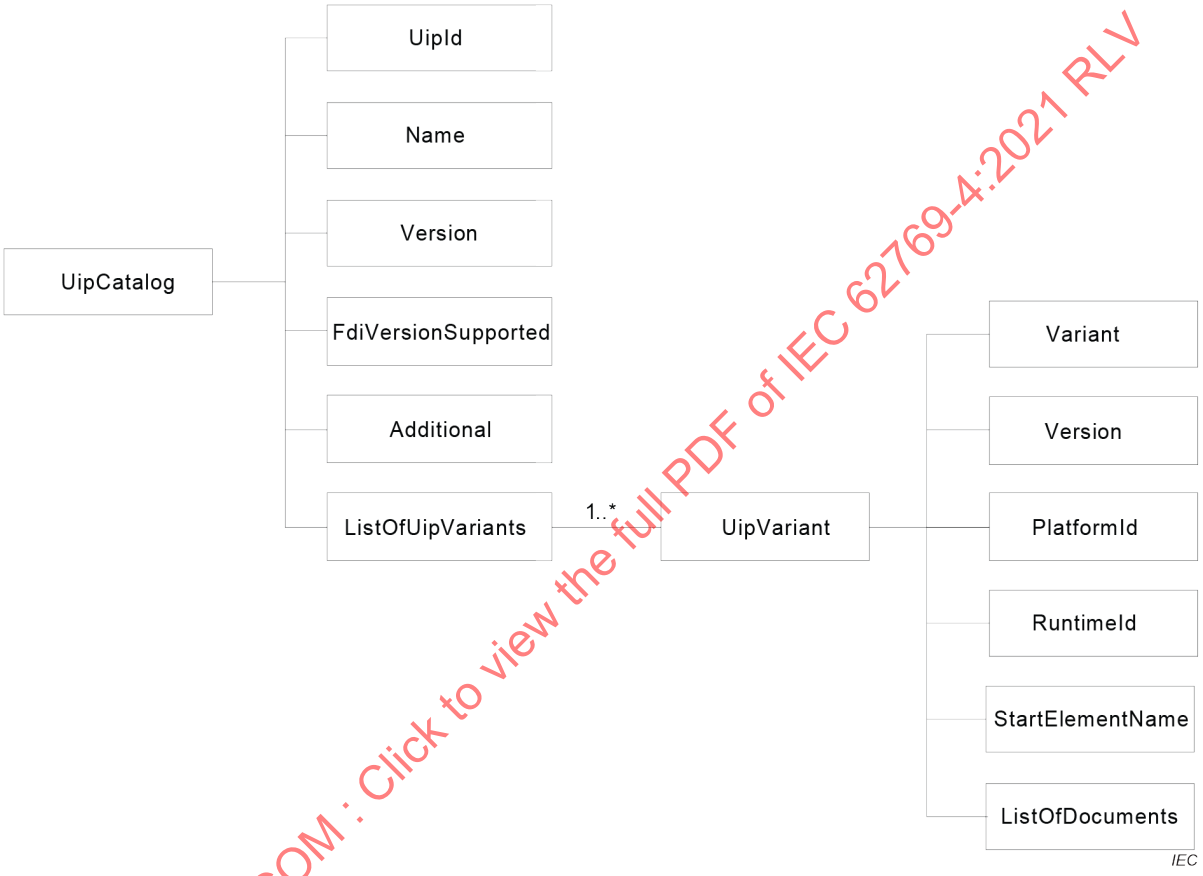

Figure 13 – Catalogue d'UIP

La partie Catalogue d'UIP est décrite dans le Tableau 5.

Tableau 5 – Partie de Catalogue d'UIP

Partie	Contenu
Type de contenu	application/vnd.fdi.package.uip.catalog+xml
Espace de noms racine	http://fdi-cooperation.com/2010/uip-catalog
Relation source	http://fdi-cooperation.com/2010/relationships/uip-catalog
Nom de fichier	uipcatalog.xml

5.3.3.2.2.1.2 Contenu

Le schéma pour le Catalogue d'UIP est décrit dans l'Annex E.

5.3.3.2.2.2 Variante d'UIP

5.3.3.2.2.2.1 Format

Une variante d'UIP peut être fournie pour prendre en charge différentes versions de la technologie de base. Il s'agit d'un conteneur pour tous les fichiers physiques qui représentent la Variante d'UIP, indépendamment de leur type et de leur nombre. La Variante d'UIP doit utiliser le format de fichier ZIP (archive compressée, type de support application/zip) comme technologie de groupage.

L'utilisation de la spécification ZIP pour les variantes de groupage doit être limitée aux exigences indiquées dans l'ISO/IEC 29500-2:2016, Annexe C.

La partie de la Variante d'UIP est décrite dans le Tableau 6.

Tableau 6 – Partie Variante d'UIP

Partie	Contenu
Type de contenu	application/zip
Espace de noms racine	Non applicable
Relation source	http://fdi-cooperation.com/2010/relationships/uip-variant
Nom de fichier	Non spécifié

5.3.3.2.2.2.2 Contenu

Le contenu de la variante d'UIP est spécifié dans l'IEC 62769-6.

5.3.4 Pièces jointes

5.3.4.1 Image

Une image en pièce jointe du Paquetage FDI est une icône de l'ordinateur qui représente le type d'appareil. Plusieurs images et résolutions du type d'appareil sont prises en charge. Chaque image doit être au format PNG et limitée aux résolutions en pixels suivantes: 256 × 256, 64 × 64, 32 × 32 ou 16 × 16. La partie Image est décrite dans le Tableau 7.

Si le Paquetage d'Appareil est réputé prendre en charge les appareils portables, l'image fournie doit être disponible dans une résolution de 16 pixels × 16 pixels.

Tableau 7 – Partie Image

Partie	Contenu
Type de contenu	image/png
Espace de noms racine	Non applicable
Relation source	http://fdi-cooperation.com/2010/relationships/attachment-image
Nom de fichier	Non spécifié

5.3.4.2 Documentation

Les parties de documentation comprennent les documents tels que les manuels de produits et les fiches techniques et doivent être codées au format PDF ou texte en clair. La Pièce jointe de la documentation est décrite dans le Tableau 8.

Tableau 8 – Partie de documentation

Partie	Contenu
Type de contenu	Application/pdf (spécifié dans l'ISO 32000-1) Text/plain (texte en clair)
Espace de noms racine	Non applicable
Relation source	http://fdi-cooperation.com/2010/relationships/attachment-documentation
Nom de fichier	Non spécifié

5.3.4.3 Fichier de prise en charge du protocole

La partie fichier de prise en charge du protocole est un fichier qui n'est pas défini par le présent document, mais qui est nécessaire pour intégrer le produit dans un système. Ces fichiers étendent habituellement le mécanisme d'intégration fourni dans la partie fondamentale du Paquetage d'Appareil FDI (EDD, UIP) par le biais de moyens spécifiques au protocole.

Ces fichiers ne doivent pas être des exécutables ni des fichiers binaires d'aucune sorte. En outre, ces fichiers ne doivent pas remplacer des éléments obligatoires d'un Paquetage d'Appareil FDI.

Une liste des fichiers de prise en charge de protocole relatifs aux différents protocoles de communication est définie à l'Annex F. La partie fichier de prise en charge spécifique à un protocole est décrite dans le Tableau 9.

Tableau 9 – Partie Fichier de Prise en charge du Protocole

Partie	Contenu
Type de contenu	Non spécifié ici
Espace de noms racine	Non spécifié ici
Relation source	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Nom de fichier	Non spécifié ici

5.3.4.4 Certificat d'enregistrement de FDI

5.3.4.4.1 Format

Le Certificat d'enregistrement de FDI est un document XML unique par Paquetage FDI qui peut être présent (élément de Paquetage FDI facultatif). Dans le domaine d'application d'un Paquetage FDI, il s'agit d'un Élément de Paquetage du type Attachment (Pièce jointe).

Tableau 10 – Partie Certificat d'enregistrement de FDI

Partie	Contenu
Type de contenu	application/vnd.fdi.package.registrationCert+xml
Espace de noms racine	Non spécifié ici
Relation source	http://fdi-cooperation.com/2010/relationships/attachment-registrationCert
Nom de fichier	RegistrationCert.xml

5.3.4.4.2 Contenu

Le Certificat d'Enregistrement de FDI lisible par la machine est représenté au format XML et son contenu et sa structure sont conformes à la Figure 14. Le Certificat d'Enregistrement de FDI peut être signé avec une signature détachée dans le même document.

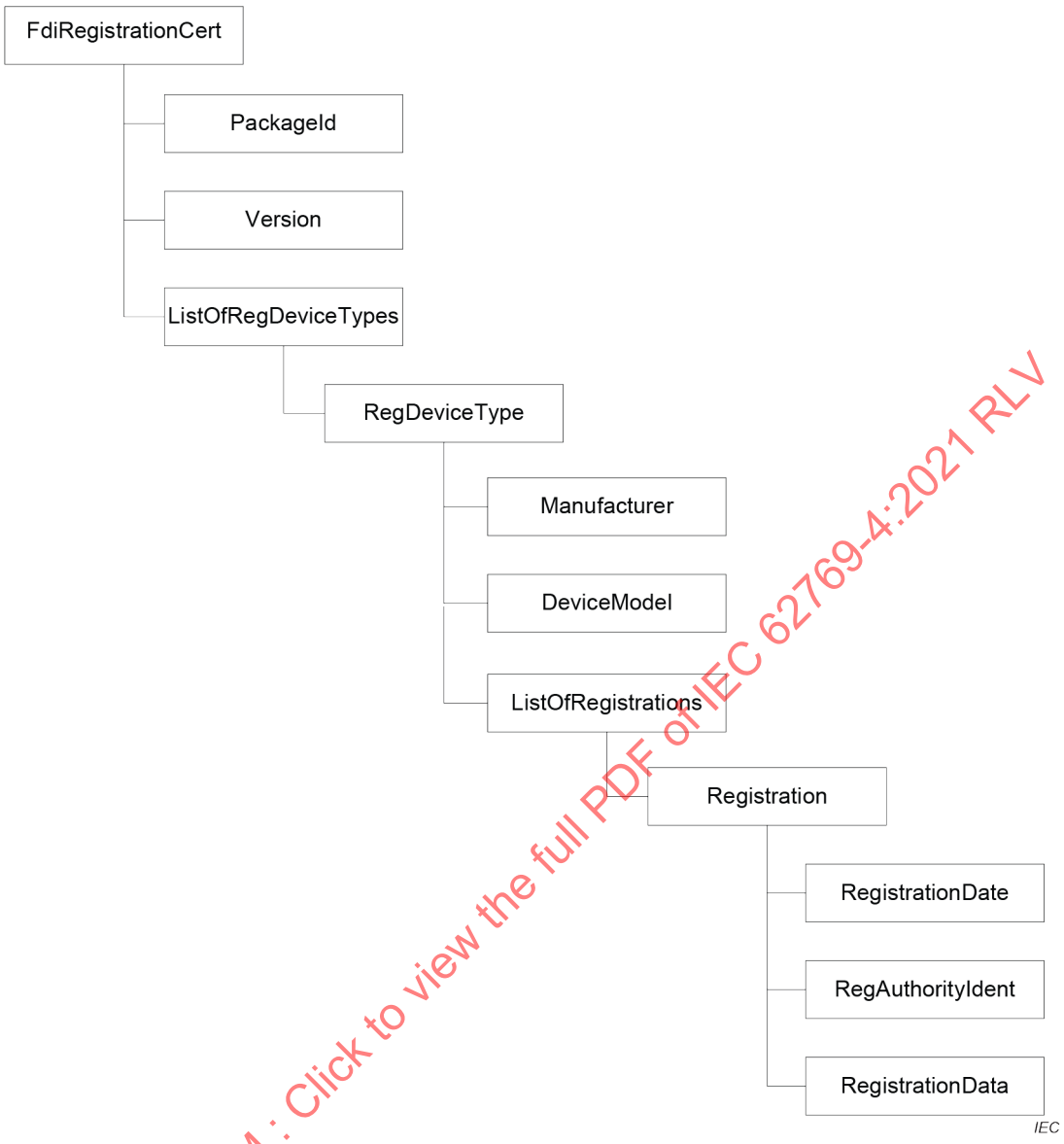

Figure 14 – Certificat d'Enregistrement de FDI

Le schéma pour le Certificat d'Enregistrement FDI est décrit à l'Annex E.

6 Gestion de versions du Paquetage FDI

6.1 Schéma de version

Les éléments FDI utilisent un schéma numérique de gestion de versions "majeure.mineure.révision" pour le paquetage et les éléments du paquetage. La version initiale doit être 1.0.0.

Les modifications incompatibles sont indiquées par incrémentation du numéro majeur. Les modifications fonctionnelles qui maintiennent encore la compatibilité avec la version majeure sont indiquées par incrémentation du numéro mineur. Les modifications non fonctionnelles, telles que les modifications rédactionnelles, sont indiquées par l'incrémentation du numéro de révision. Les règles utilisées pour incrémenter le numéro de version sont décrites en 6.3.

Des exemples de méthode de déploiement de différents types de paquetage FDI sont décrits à l'Annex C.

6.2 Éléments versionnés

Un Paquetage FDI contient les informations relatives à la version qui sont décrites dans le Tableau 11.

Tableau 11 – Éléments versionnés

Domaine d'application	Emplacement des versions	Références du schéma
Paquetage FDI (voir 5.3)	Catalogue de Paquetage (voir 5.3.1)	Voir l'Annex E, élément de version de type complexe Package
UIP (voir 5.3.3)	Catalogue d'UIP (voir 5.3.3.2.2.1)	Voir l'Annex E, élément de version de type complexe Uip
Variante d'UIP (voir 5.3.3.2.2.2)	Catalogue d'UIP (voir 5.3.3.2.2.1)	Voir l'Annex E, élément de version de type complexe UipVariant

Les parties suivantes du Paquetage FDI ont des mécanismes de version qui ne relèvent pas du domaine d'application de la FDI.

- EDD (voir 5.3.2)
  - DD_REVISION et DEVICE_REVISION conformément à l'IEC 61804-3
- Pièces jointes (voir 5.3.4)
  - Image (voir 5.3.4.1)
  - Documentation (voir 5.3.4.2)
  - Fichier de prise en charge du protocole (voir 5.3.4.3 et l'Annex F)

6.3 Hiérarchie des Versions

Le changement de version des éléments du Paquetage FDI peut influencer la version d'éléments qui se chevauchent. La Figure 15 représente ces influences. Les rectangles orange représentent des éléments qui sont versionnés en utilisant les mécanismes de versions spécifiques de FDI décrits dans l'IEC 62769-1. Les rectangles verts représentent les éléments qui ont des mécanismes de versions qui ne sont pas décrits par FDI. Les rectangles blancs représentent les éléments qui ne sont pas explicitement versionnés.

Tous les éléments de la structure arborescente indiquée ci-dessous influencent la version des éléments parents selon le Tableau 12. Leur modification de version provoque une modification de version des éléments qui se chevauchent.

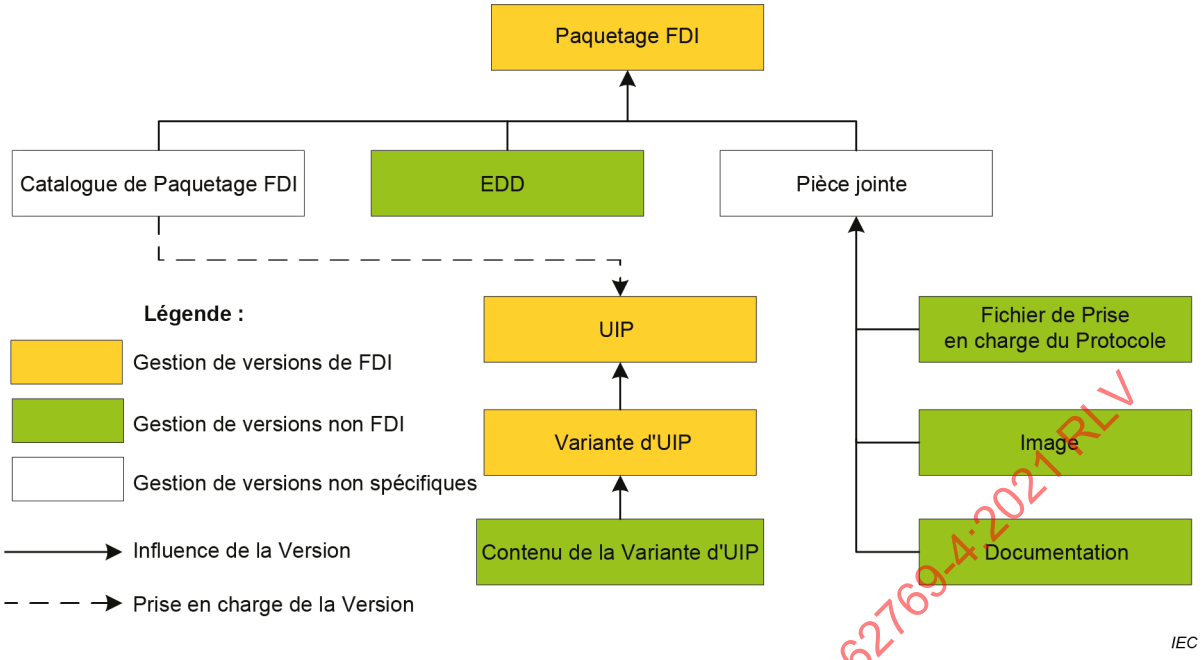

Figure 15 – Hiérarchie des Versions

Le Tableau 12 décrit une sélection de modifications possibles des différents éléments de Paquetage FDI et leur influence directe sur la version de Paquetage FDI. Au minimum, les versions successives de Paquetage FDI doivent incrémenter le numéro de révision.

Tableau 12 – Influence sur la Version de Paquetage FDI

Élément	Niveau de Version			Type de Paquetage FDI	
	Majeure	Mineure	Révision	Appareil/ Profil/ Paquetage de Comm	Paquetage d'UIP
Catalogue de Paquetage	n/a	Modification des références de la compatibilité existante. Modification de ListOfSupportedUips (voir Annex E). L'ajout de caractères génériques n'est permis que pour les modifications de VersionSupported existante (voir Annex E)	Rédactionnelle	x	x
EDD	Incrémentation des Révisions de l'Appareil	Incrémenter DD_REVISION avec le même DEVICE_REVISION	n/a	x	
Pièces jointes	Modification des Fichiers de Prise en charge du Protocole	Modification des Fichiers de Prise en charge du Protocole	Modification de la Documentation, des Images, des Fichiers de Prise en charge du Protocole	x	
UIP	Modifications incompatibles Axé sur la technologie	Amélioration fonctionnelle Axé sur la technologie	Correction du bogue Axé sur la technologie	x	x

L'Annex G décrit les cas d'utilisation typiques du cycle de vie de Paquetage FDI pour une meilleure compréhension de la hiérarchie des versions et du concept de gestion de versions.

6.4 Compatibilité de l'UIP

Les UIP sont inclus dans un Paquetage d'Appareil FDI, un Paquetage de Communication FDI, un Paquetage de Profil FDI ou un Paquetage d'UIP FDI. Un UIP inclus dans un Paquetage d'UIP FDI ne doit être inclus dans aucun autre type de paquetage. Par conséquent, un Paquetage FDI peut ne pas être autonome.

Les Paquetages d'Appareil FDI, les Paquetages de Communication FDI et les Paquetages de Profil FDI ne référencent pas directement les UIP. En revanche, le Catalogue de Paquetage définit SupportedUip (voir l'Annex E) pour définir les UIP compatibles. Plusieurs UIP installés peuvent être compatibles avec une version du Paquetage FDI donné. Cette disposition permet aux créateurs de Paquetage FDI de corriger les bogues et d'apporter des améliorations fonctionnelles au Paquetage FDI sans qu'il soit nécessaire de fournir une version majeure d'un Paquetage FDI.

La prise en charge de la version décrite dans le Catalogue de Paquetage est définie comme suit:

- Le numéro de la version majeure doit être spécifié explicitement comme un seul numéro.
- Le numéro de la version mineure et le numéro de révision doivent être spécifiés explicitement comme un seul numéro ou comme un caractère générique ('*'). Si le numéro de la version mineure est spécifié comme un caractère générique, le numéro de révision doit également être spécifié comme un caractère générique.

NOTE Les exemples de numéros de versions compatibles valides sont 1.3.1, 1.3.*, 1.*.*.

L'utilisation d'un caractère générique indique que tous les numéros mineurs et/ou les numéros de révision sont compatibles avec le Paquetage FDI. Si le SupportedUip a été défini à l'aide de caractères génériques et s'il existe plusieurs versions d'UIP disponibles, le Serveur FDI doit transférer la dernière version au Client FDI. Les mises en œuvre spécifiques au système qui permettent la coexistence d'une version mineure ou des versions de révisions indépendamment des caractères génériques de prise en charge de la version ne relèvent pas du domaine d'application du présent document.

La Figure 16 représente un exemple de sélection de l'UIP compatible le plus récent.

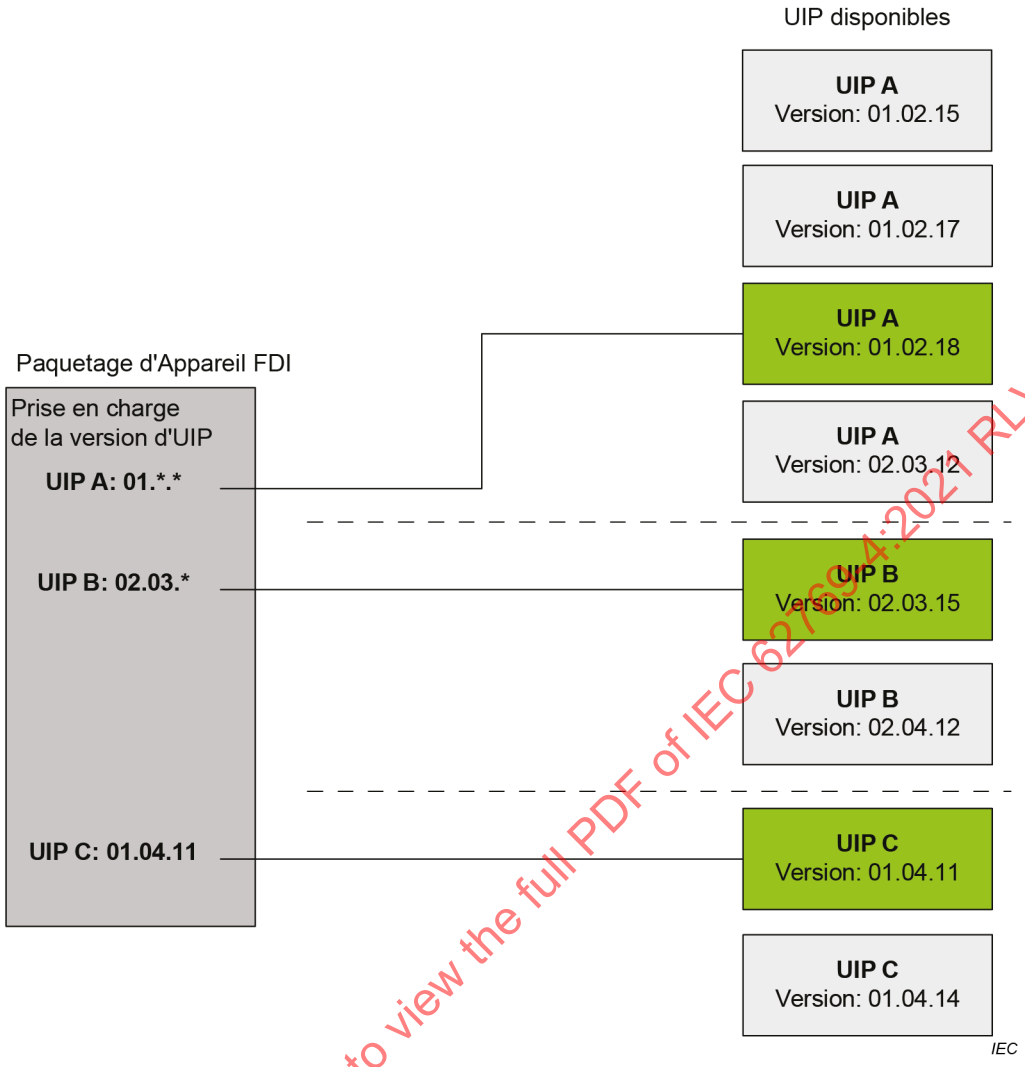

Figure 16 – Concept de Prise en charge de la Version de l'UIP

7 Signatures numériques et Certificats d'enregistrement

7.1 Éléments signés et documents de certification

La politique de signature du Paquetage FDI stipule que l'ensemble de Paquetage FDI doit être signé par l'émetteur du Paquetage FDI (voir 7.3). Il convient que les Paquetages FDI qui ont été enregistrés par une Autorité d'enregistrement de FDI comportent un Certificat d'enregistrement de FDI (pièce jointe spéciale, voir 5.3.4.4), qui doit être signé par une Autorité d'enregistrement de FDI (voir 7.3). Les Paquetages FDI sont enregistrés s'ils ont satisfait aux essais de conformité (voir B.2.4).

Il convient que les Paquetages FDI enregistrés comportent un ou plusieurs Certificats d'enregistrement de FDI signés numériquement pour

– indiquer que les Paquetages FDI ont été enregistrés par une autorité d'enregistrement officielle de FDI,

– indiquer que les fichiers du paquetage, qui ont été critiques pour l'enregistrement du paquetage, n'ont pas été modifiés après soumission de l'enregistrement.

Les Paquetages FDI dans leur ensemble (l'entité environnante couvrant tous les éléments) doivent être signés pour

– identifier l'émetteur (le signataire) du Paquetage FDI;
– vérifier que le Paquetage FDI signé n'a pas été altéré après l'apposition de la signature.

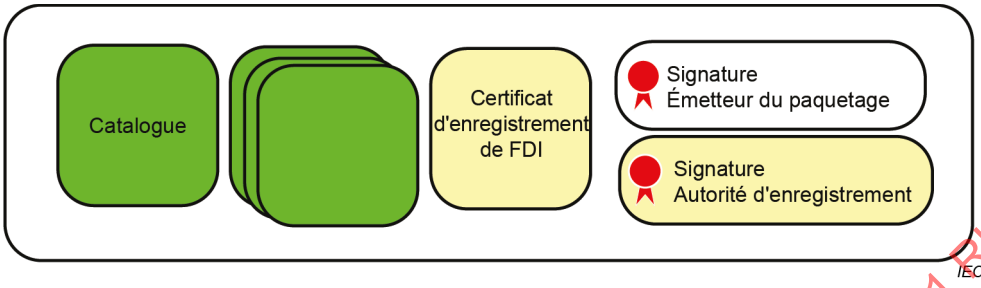

Figure 17 – Signature du Paquetage FDI

L'émetteur du Paquetage FDI envoie tout d'abord un paquetage aux Autorités d'enregistrement de FDI. Ces dernières effectuent des essais de conformité définis et des essais supplémentaires selon leurs règles spécifiques et leurs descriptions d'essai ou leurs accords.

Après des essais satisfaisants, ces Autorités d'enregistrement peuvent délivrer un Certificat d'enregistrement de FDI qui doit être signé par l'Autorité d'enregistrement FDI individuelle. Le Certificat d'enregistrement de FDI peut contenir plusieurs informations relatives à l'enregistrement telles que, un hachage sur les parties du Paquetage FDI qui ont fait l'objet des essais de conformité. Le Certificat d'enregistrement de FDI est intégré dans le Paquetage FDI pour publication. Après ajout du Certificat d'enregistrement de FDI au contenu du Paquetage FDI, le paquetage est de nouveau signé et ensuite émis/publié par l'émetteur.

L'émetteur du Paquetage FDI confirme donc que le Certificat d'enregistrement de FDI est fiable en signant l'ensemble du Paquetage FDI qu'il publie.

7.2 Mécanisme de signature

Toutes les signatures au sein du Paquetage FDI doivent être apposées conformément au mécanisme défini dans l'ISO/IEC 29500-2. En plus de satisfaire aux exigences de l'ISO/IEC 29500-2, les signatures doivent satisfaire aux exigences suivantes:

– Les informations nécessaires à la validation de la signature doivent faire partie de la signature numérique, c'est-à-dire que l'élément KeyInfo spécifié dans la Syntaxe et le Traitement XML de Signature est obligatoire.
– Les Certificats utilisés pour la signature doivent être associés à une racine dans une Autorité de certification incluse dans les Autorités de certification de confiance du Magasin de Certificats de Microsoft Windows[1].
– Les algorithmes utilisés pour la création de signatures (pour le hachage et le codage/décodage) doivent provenir de la liste des algorithmes recommandés par le NIST dans le document FIPS 140-2, Annexe A (NIST).
– La signature doit inclure un horodatage de confiance conformément à XAdES (XML Advanced Electronic Signatures (Signatures électroniques avancées XML – ETSI EN 319 132-1).

[1] Microsoft Windows Certificate Store est l'appellation commerciale d'un produit distribué par Microsoft®.Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit cité. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

– Toute signature doit inclure une CommitmentTypeIndication (Indication du type d'engagement) conformément à l'ETSI TS 101 733. Les types d'engagements utilisés sont spécifiés en 7.3.

7.3 Émetteur du Paquetage FDI, Autorité d'enregistrement de FDI

L'Émetteur du Paquetage FDI et l'Autorité d'enregistrement de FDI ont les responsabilités suivantes:

– Un émetteur du Paquetage FDI publie officiellement un Paquetage FDI et le signe pour assurer l'intégrité du Paquetage FDI. Le Paquetage FDI peut être créé par un fournisseur d'appareils ou un fournisseur de solutions logicielles. L'éditeur d'un Paquetage FDI peut être une personne différente. Le type d'engagement est ProofOfOrigin.

– Une Autorité d'enregistrement de FDI a le droit et la capacité d'effectuer des essais de conformité FDI sur des Paquetages FDI et de délivrer des Certificats d'enregistrement de FDI. Il s'agit en général de groupes d'intérêt représentant un protocole de communication pris en charge par FDI ou leurs partenaires agréés. Le type d'engagement est ProofOfApproval.

7.4 Comportement de l'Hôte FDI

Un système Hôte FDI doit afficher un message d'avertissement lorsque la procédure d'importation du Paquetage FDI reconnaît que:

– une signature numérique sur le paquetage n'est pas présente ou n'inclut pas toutes les entités (fichiers) à l'intérieur du paquetage;

– la signature numérique en tant que telle n'est pas digne de confiance;

– la signature est altérée, ce qui indique que le paquetage a été modifié après la signature.

Les mesures supplémentaires de sécurité à prendre, lorsque le message d'avertissement a été affiché, relèvent de la responsabilité du système Hôte FDI.

Il convient qu'un système Hôte FDI affiche un message d'information qui indique les parties dont l'enregistrement a été modifié lorsque la procédure d'importation du Paquetage FDI reconnaît que

– l'identificateur unique (PackageID) et la version (telle que définie à l'Annex E) du Paquetage FDI ne correspondent pas aux informations fournies dans le fichier de Certificat d'enregistrement de FDI;

– aucun Certificat d'enregistrement de FDI n'est présent dans le Paquetage FDI;

– le Certificat d'enregistrement de FDI inclus n'est pas signé, la signature n'est pas digne de confiance ou la signature est altérée.

Un système Hôte FDI peut vérifier la signature et le statut de certification en lisant le Certificat d'enregistrement de FDI. Un Hôte doit fournir une configuration qui permet d'importer un Paquetage FDI qui ne comprend pas de Certificat d'enregistrement de FDI. La fonctionnalité de ce Paquetage FDI ne doit pas être limitée.

Annex A
(normative)

Conventions relatives aux noms de fichiers

A.1 Identification

L'identification des conventions de dénomination doit être utilisée pour fournir un moyen unique d'identification des Paquetages FDI complets ainsi que des éléments de Paquetages FDI. Le respect des règles d'identification favorise l'interopérabilité entre les systèmes. Toutefois, les noms eux-mêmes ne doivent pas être le seul mécanisme pour le déploiement.

Habituellement, les conventions de dénomination lisibles par une machine ont été utilisées pour lier de manière unique un fichier à un appareil spécifique et faciliter l'importation et l'utilisation des fichiers de l'appareil d'interface, tel qu'une EDD, de sorte qu'il devient difficile pour les utilisateurs de déterminer si le fichier exigé était disponible dans le système de fichiers. Étant donné qu'un Paquetage FDI est l'élément visible pour l'utilisateur, un format lisible par l'homme est préférentiel par rapport à celui d'une version lisible par une machine.

A.2 Convention relative aux noms de fichier du Paquetage FDI

Les Paquetages FDI individuels doivent être identifiés par des noms de fichiers uniques, qui doivent être composés de la fabrication, du modèle ou type, de la révision et du protocole pris en charge. En raison de la persistance des problèmes liés au nom de fichier, celui-ci ne doit pas être le seul moyen d'identifier un Paquetage FDI. Des moyens sécurisés d'identification d'un Paquetage FDI doivent être utilisés pour empêcher le mélange inapproprié de l'appareil avec les Paquetages FDI.

Le Paquetage FDI doit utiliser la convention de dénomination suivante:

<manufacturer>.<description>.<major>.<minor>.<revision>.<protocol>.fdix
(<fabricant>.<description>.<majeure>.<mineure>.<révision>.<protocole>.fdix)

Chaque élément du nom de fichier est décrit dans le Tableau A.1.

Les règles suivantes doivent s'appliquer pour dénommer un Paquetage FDI:

- Tous les Paquetages FDI doivent utiliser l'extension .fdix.
- Longueur maximale du nom qui contient l'extension .fdix: 128 caractères.
- Le nom de fichier ne doit pas inclure d'espace.

NOTE La longueur maximale d'un nom de fichier est limitée à 128 caractères afin d'éviter que le chemin ne dépasse la valeur maximale.

Tableau A.1 – Convention de Dénomination du Paquetage FDI

Élément de nom de fichier	Description
fabricant	Chaîne de caractères représentant le nom du fabricant de l'appareil. Le fabricant ne doit pas inclure de point.
description	Description sommaire du paquetage. La description ne doit pas inclure de point. Pour un Paquetage d'Appareil FDI, chaîne de caractères représentant le nom du type d'appareil.
majeure	Représentation numérique de deux caractères de la version majeure du Paquetage FDI.
mineure	Représentation numérique de deux caractères de la version mineure du Paquetage FDI.
révision	Représentation numérique de deux caractères de la révision du Paquetage FDI.
protocole	Les noms de famille de profils de communication sont définis dans les profils de communication (IEC 62769-1xx-x).

Annex B
(informative)

Création d'un Paquetage FDI

B.1 Généralités

L'Annex B décrit les principes fondamentaux d'un possible processus de création de Paquetage FDI à l'aide d'outils et de composants de développement normalisés disponibles.

B.2 Outils et composants

B.2.1 Vue d'ensemble

Les Paquetages FDI sont constitués de plusieurs composants différents, qui peuvent avoir également des relations et des dépendances entre eux, sont développés au moyen de différentes technologies de mise en œuvre, et qui respectent différentes normes. Cette structure complexe de Paquetages FDI exige un outil qui assure un développement et une maintenance économiques et aisés.

B.2.2 Mise en œuvre de Référence de FDI / Moteur EDD Commun

La Mise en œuvre de Référence de FDI, qui comporte un Moteur EDD Commun, vérifie qu'une mise en œuvre commune est utilisée pour la mise en œuvre du Paquetage FDI et de l'essai qui a un comportement défini. La Mise en œuvre de Référence de FDI fait partie intégrante des deux outils mentionnés ci-dessous.

B.2.3 Environnement de développement Intégré (IDE) du Paquetage FDI

L'IDE du Paquetage FDI fournit tout ce qui est nécessaire pour gérer des projets de développement pour les différents types de Paquetages FDI, le développement des parties descriptives, la liaison de toutes les parties restantes du paquetage, mais également pour assurer l'empaquetage réel du paquetage.

B.2.4 Outil d'essai de conformité du Paquetage d'Appareil FDI

Un Paquetage FDI développé et en particulier les Paquetages d'Appareil FDI sont soumis à l'essai pour établir la conformité de la mise en œuvre de la spécification FDI, ce qui assure une certaine interopérabilité. L'essai de conformité est effectué en utilisant l'outil d'essai de conformité du Paquetage d'Appareil FDI qui exécute les cas d'essai définis avec la Mise en œuvre de Référence FDI et le Paquetage FDI développé.

B.3 Développement

B.3.1 Développement de base du Paquetage FDI

L'IDE du Paquetage FDI permet la création d'un projet de développement pour le Paquetage FDI, y compris le type de projet et les informations relatives à la version. Ce projet de développement peut être utilisé pendant le cycle de vie complet du Paquetage FDI. Un assistant de projet de développement peut être disponible afin d'accélérer le processus de création du projet.

Un composant d'éditeur qui accompagne l'IDE du Paquetage FDI est disponible pour mettre en œuvre la partie descriptive (EDD) du Paquetage FDI. Des fonctionnalités telles que les vérifications de syntaxe, le pliage de code, l'autocomplétion et les assistants pour les constructions complexes sont disponibles pour prendre en charge un développement efficace et sûr.