INTERNATIONAL STANDARD

ISO 15745-4

First edition
2003-11-15

Industrial automation systems and integration — Open systems application integration framework —

Part 4: Reference description for Ethernet-based control systems

Systèmes d'automatisation industrielle et intégration — Cadres d'intégration d'application pour les systèmes ouverts —

Partie 4: Description de référence pour les systèmes de contrôle fondés sur Ethernet

Reference number
ISO 15745-4:2003(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 15745 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 15745-4 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*.

ISO 15745 consists of the following parts, under the general title *Industrial automation systems and integration — Open systems application integration framework*:

— *Part 1: Generic reference description*

— *Part 2: Reference description for ISO 11898-based control systems*

— *Part 3: Reference description for IEC 61158-based control systems*

— *Part 4: Reference description for Ethernet-based control systems*

Introduction

The application integration framework (AIF) described in ISO 15745 defines elements and rules that facilitate:

— the systematic organization and representation of the application integration requirements using integration models;

— the development of interface specifications in the form of application interoperability profiles (AIPs) that enable both the selection of suitable resources and the documentation of the "as built" application.

ISO 15745-1 defines the generic elements and rules for describing integration models and AIPs, together with their component profiles - process profiles, information exchange profiles, and resource profiles. The context of ISO 15745 and a structural overview of the constituents of an AIP are given in Figure 1 of ISO 15745-1:2003.

This part of ISO 15745 extends the generic AIF described in ISO 15745-1 by defining the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles specific to Ethernet[1]-based control systems (ADS-net[2], FL-net[3] and EtherNet/IP™[4]). EtherNet/IP™ technology uses a profile of IEC 61158 which is specified in IEC 61784-1.

In particular, this part of ISO 15745 describes technology specific profile templates for the device profile and the communication network profile. Within an AIP, a device profile instance or a communication network profile instance is part of the resource profile defined in ISO 15745-1. The device profile and the communication network profile XML instance files are included in a resource profile XML instance using the ProfileHandle_DataType as specified in ISO 15745-1:2003, 7.2.5.

AIFs specified using the elements and rules of ISO 15745-1 can be easily integrated with the component profiles defined using the elements and rules specified in this part.

---

[1] Ethernet is used in this document as a synonym for ISO/IEC 8802-3.

[2] ADS-net is a trade name used to describe JIS-TR B0012 (Autonomous Decentralized System network). This information is given for the convenience of users of ISO 15745 and does not constitute an endorsement by ISO of the trademark, or any related products. Compliance to this standard does not require use of the trade name ADS-net.

[3] FL-net is a trade name used to describe JEM 1479. This information is given for the convenience of users of ISO 15745 and does not constitute an endorsement by ISO of the trademark, or any related products. Compliance to this standard does not require use of the trade name FL-net.

[4] EtherNet/IP™ is a trade name of ControlNet International, Ltd. and Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of ISO 15745 and does not constitute an endorsement by ISO of the trademark holder or any of its products. Compliance to this standard does not require use of the trade name EtherNet/IP™. Use of the trade name EtherNet/IP™ requires permission of either ControlNet International, Ltd. or Open DeviceNet Vendor Association, Inc.

Industrial automation systems and integration — Open systems application integration framework —

Part 4: Reference description for Ethernet-based control systems

1 Scope

This part of ISO 15745 defines the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles specific to Ethernet-based control systems.

NOTE Generic elements and rules for describing integration models and application interoperability profiles, together with their component profiles (process profiles, information exchange profiles, and resource profiles) are specified in ISO 15745-1.

This part of ISO 15745 is to be used in conjunction with ISO 15745-1 to describe an application integration framework.

2 Normative references

The following referenced documents are indispensable for the application 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.

ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code*

ISO 15745-1:2003, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework*

ISO/IEC 8802-3:2000, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

IEC 61158 (all parts), *Digital data communications for measurement and control – Fieldbus for use in industrial control systems*

IEC 61784-1:2003, *Digital data communications for measurement and control – Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems*

IEEE Std 754-1985 (R1990), *IEEE Standard for Binary Floating Point Arithmetic*

JEM 1479:2002, *Protocol specification – Factory automation control Link Network (FL-net)*

JIS-TR B0012: 2000, *Autonomous Decentralized System Network (ADS-net)*

REC-xml-20001006, *Extensible Markup Language (XML) 1.0 Second Edition – W3C Recommendation 6 October 2000*

REC-xmlschema-1-20010502, *XML Schema Part 1: Structures – W3C Recommendation 02 May 2001*

REC-xmlschema-2-20010502, *XML Schema Part 2: Datatypes – W3C Recommendation 02 May 2001*

RFC 768:1980, *User Datagram Protocol – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

RFC 791:1981, *Internet Protocol – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

RFC 793:1981, *Transmission Control Protocol – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

RFC 894:1984, *A standard for the Transmission of IP Datagrams over Ethernet Networks – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

RFC 1738:1994, *Uniform Resource Locators (URL) – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

RFC 1759:1995, *Printer MIB – Internet Engineering Task Force (IETF), Request for Comments (RFC)*

UML V1.4, *OMG - Unified Modeling Language Specification (Version 1.4, September 2001)*

3 Terms and definitions

NOTE The UML terminology and notation used in this document is described in Annex A of ISO 15745-1:2003.

For the purposes of this document, the terms and definitions given in ISO 15745-1 apply.

4 Abbreviated terms

AIF	Application Integration Framework
AIP	Application Interoperability Profile
ASCII	American Standard Code for Information Interchange
CIP™[5]	Common Industrial Protocol
EDS	Electronic Data Sheet
IAS	Industrial Automation Systems
IP	Internet Protocol (see RFC 791 and RFC 894)
MAC	Media Access Control
MCG	MultiCast Group

[5] CIP™ is a trade name of ControlNet International, Ltd. and Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of ISO 15745 and does not constitute an endorsement by ISO of the trademark holder or any of its products. Compliance to this standard does not require use of the trade name CIP™. Use of the trade name CIP™ requires permission of either ControlNet International, Ltd. or Open DeviceNet Vendor Association, Inc.

OSI	Open System Interconnection
TCD	Transaction CoDe
TCP	Transmission Control Protocol (see RFC 793)
UDP	User Datagram Protocol (see RFC 768)
UML	Unified Modeling Language (see UML V1.4)
XML	eXtensible Markup Language (see REC-xml-20001006)

5 Technology specific elements and rules

5.1 Integration models and IAS interfaces

The AIP developer shall develop the integration model using the rules described in ISO 15745-1, and shall ensure that the Ethernet-based device and communication network profiles (whether representing the interface requirements or those derived from existing devices/communication networks) include the necessary IAS interfaces. The IAS interfaces included in the profile shall be identified in the header section (see ISO 15745-1:2003, 7.2.2).

NOTE IAS interfaces are described in ISO 15745-1:2003, Annex B.

5.2 Profile templates

5.2.1 General

The Ethernet based technology specific profile templates are derived from the generic profile templates specified in ISO 15745-1:2003, clause 7.

5.2.2 Contents and syntax

ISO 15745 specifies profile templates that are XML schemas (REC-xmlschema-1-20010502 and REC-xmlschema-2-20010502) and use a common general structure. The device and communication network profiles based on these templates typically contain:

— information needed to identify the connected device,

— a description of device data that can be accessed via the network,

— a description of the communication capabilities supported by the device,

— additional vendor-specific information.

However, some Ethernet-based technologies use specific legacy ASCII syntax. Hence, for backward compatibility, template definitions of any technology (Annex A to Annex C) include all or a relevant subset of the following:

— communication network and device profile templates, as defined in ISO 15745-1,

— ISO 15745 template to encapsulate files with legacy ASCII syntax ("wrapper"),

— legacy ASCII syntax.

5.2.3 Header

The profile template header defined in ISO 15745-1:2003, 7.2.2, is used for Ethernet technology specific profile templates. Each technology uses one or more names to identify the technology or its particular component(s) (see Table 1). The selected name shall be stored in the ProfileTechnology attribute in the header section.

Table 1 — ProfileTechnology names

ProfileTechnology name	Technology
ADS-net	ADS-net
FL-net	FL-net
EtherNet/IP	EtherNet/IP
CIP	EtherNet/IP
EDS	EtherNet/IP

5.3 Technology specific profiles

The technology specific communication network profile structure and communication related aspects of device profile structure based on Ethernet technologies are described in clause 6. The technologies included are:

— ADS-net (see 6.1)

— FL-net (see 6.2)

— EtherNet/IP (see 6.3)

The related profile template definitions are specified in Annex A to Annex C.

6 Device and communication network profiles for Ethernet-based control systems

6.1 ADS-net

6.1.1 Device profile

6.1.1.1 General

Figure 1 shows the class structure of the ADS-net device profile.

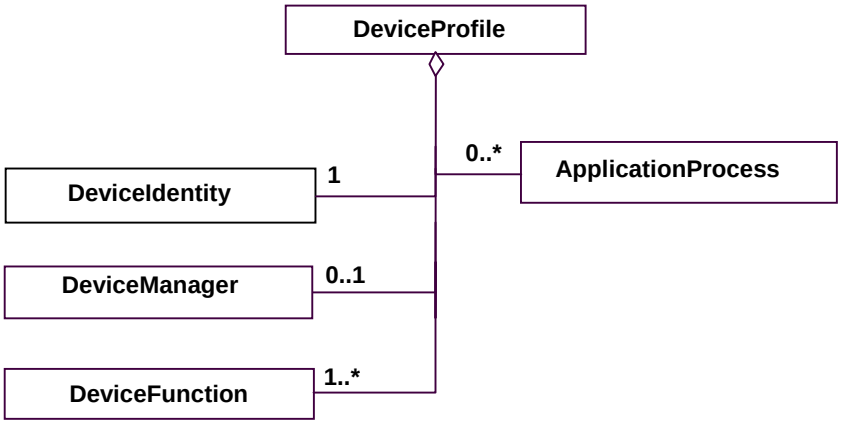

Figure 1 — ADS-net device profile class diagram

—The available formats for ADS-net device profiles are described in A.2.

The XML schema representing the ADS-net device profile template is defined in A.2.3. The file name of this XML schema shall be “ADS-net_Device_Profile.xsd”.

NOTE The ADS-net device profile class diagrams shown in Figure 1 define the main classes. These classes are further decomposed; details are defined in Annex A.

The XML schema representing the ADS-net device profile template is defined in A.2.

6.1.1.2 Device identity

The DeviceIdentity class contains attributes that enable the unique identification of the device, and supports services that enable retrieval of information from the device.

These attributes provide:

— manufacturer's identification (VendorName);

— device identification (ProductCode, ProductRevision, DeviceName).

6.1.1.3 Device manager

The DeviceManager class contains attributes and supports services that enable the monitoring and configuration of the device.

These attributes provide:

— device status (DeviceState).

6.1.1.4 Device function

The DeviceFunction class contains attributes and supports services that enable the management (e.g. configuration) of a function of the device.

These attributes provide:

— data field number (DFNO);

— logical node number (LNODENO);

— mode (MODE).

6.1.1.5 Application process

The ApplicationProcess class contains the attributes and support services that enable the communication control among application programs executed on ADS-net.

These attributes provide:

— producer TCD list (Producer-TCD-List);

— consumer TCD list (Consumer-TCD-List);

— priority (Priority).

6.1.2 Communication network profile

6.1.2.1 General

Figure 2 shows the class structure of the ADS-net communication network profile.

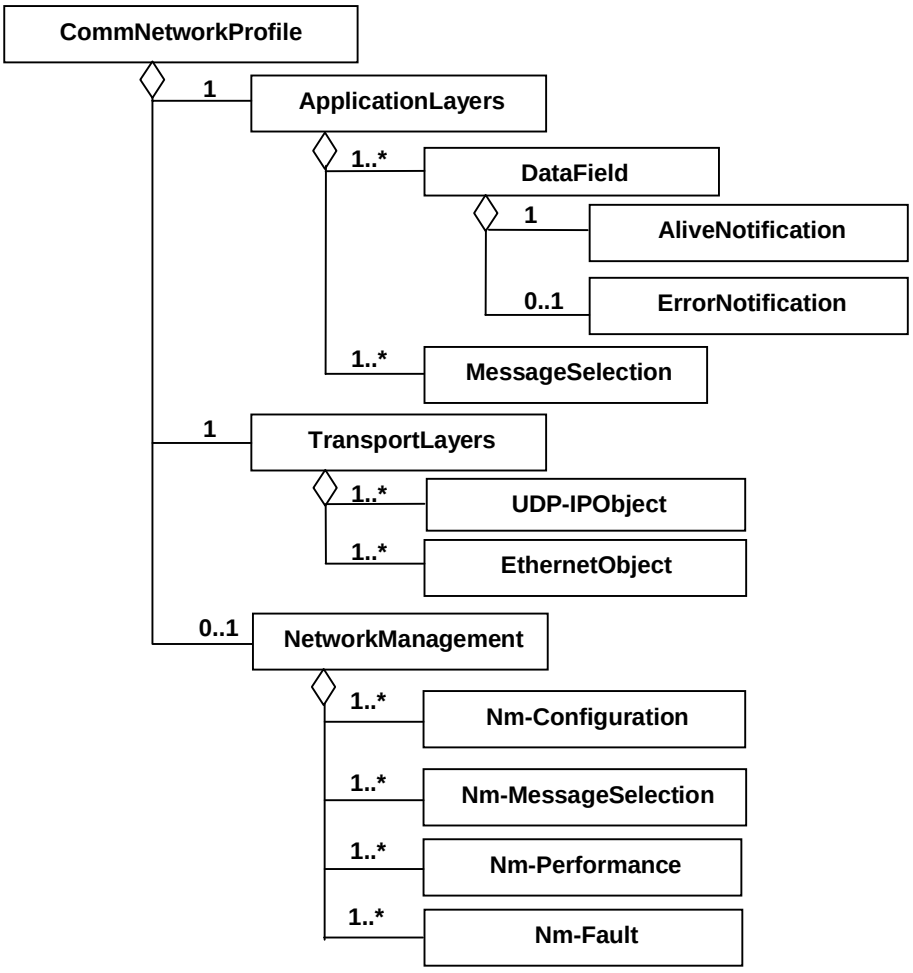

Figure 2 — ADS-net communication network profile class diagram

The available formats for ADS-net communication network profiles are described in A.3.

The XML schema representing the ADS-net communication network profile template is defined in A.3.3. The file name of this XML schema shall be “ADS-net_CommNet_Profile.xsd”.

6.1.2.2 Application layers

6.1.2.2.1 General

The ADS-net ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the ADS-net communication network integration model. It states the supported application service elements and their associated services.

Figure 3 shows the definition of the ADS-net ApplicationLayers class.

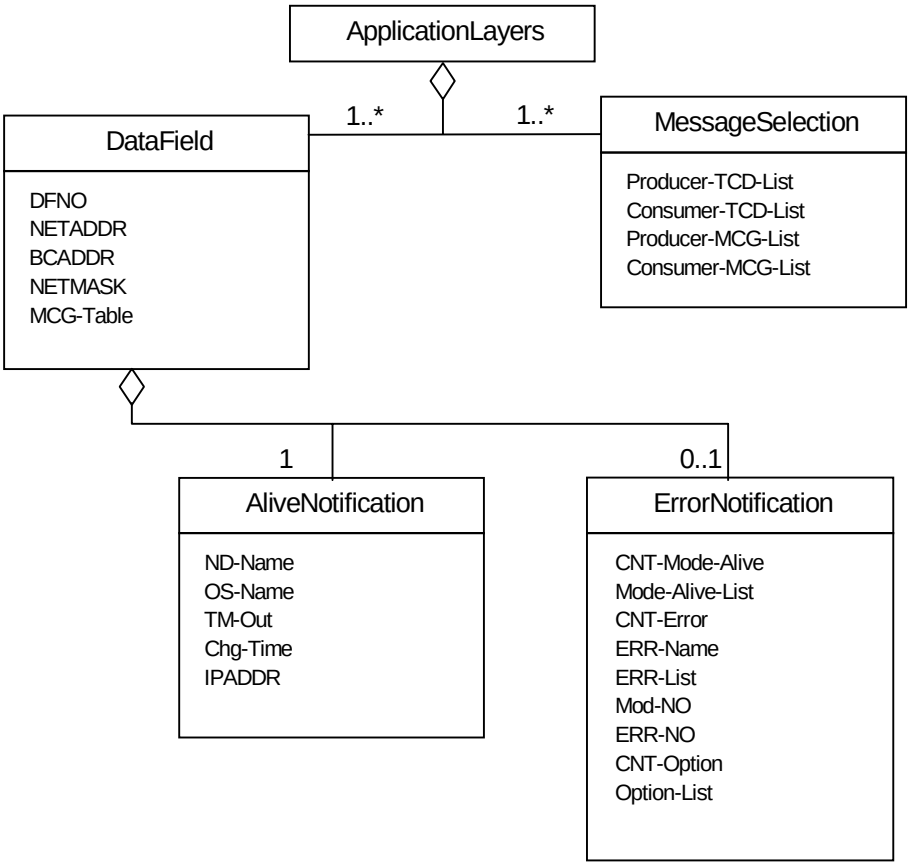

Figure 3 — ADS-net ApplicationLayers class diagram

6.1.2.2.2 DataField

6.1.2.2.2.1 General

ADS-net defines a domain named “data field” where subsystems share information by peer-to-peer message passing. Each node computer transmits messages on a multicast basis to the data field it belongs to, and the other node computers belonging to the same data field can receive the data autonomously. Multiple computers belong to a data field to send or receive data. Multicast group (MCG), a group of node computers belonging to the data field, is also introduced to localize the multicast.

DataField defines properties used to control a data field. DataField attributes provide in particular:

— data field number (DFNO);

— multicast group table (MCG-Table).

6.1.2.2.2.2 AliveNotification

A “keep alive” message is transmitted to the data field periodically to notify the status of the sender node computer to the other node computers.

AliveNotification defines the properties used to check the status of node computers. AliveNotification attributes provide in particular:

— node name (ND-Name);

— time out (TM-Out).

6.1.2.2.2.3 ErrorNotification

When a fault occurs in a node computer, the fault information is included in the “keep alive” message that is transmitted to the data field during a “keep alive” message transmission cycle. Any node computer belonging to the same data field can detect the fault (error) status.

ErrorNotification defines properties describing this fault information. ErrorNotification attributes provide in particular:

— error name (ERR-Name);

— error list (ERR-List).

6.1.2.2.3 MessageSelection

TCD is an identifier of a message defined uniquely in a data field. The transmitter sends a TCD-assigned message to a specified data field on a multicast basis, while each node computer belonging to the same data field autonomously selects only the relevant messages based on their TCD.

MessageSelection defines properties used for this message exchange. MessageSelection attributes provide in particular:

— producer TCD list (Producer-TCD-List);

— consumer TCD list (Consumer-TCD-List);

— producer MCG list (Producer-MCG-List);

— consumer MCG list (Consumer-MCG-List).

6.1.2.3 Transport layers

6.1.2.3.1 General

The ADS-net TransportLayers class represents the combined profiles for the lower 4 OSI layers of the communication network integration model. The TransportLayers class is divided into one or more Ethernet based objects and a UDP/IP object.

6.1.2.3.2 EthernetObject

EthernetObject defines properties of Ethernet used to implement ADS-net. EthernetObject attributes provide:

— media type (MediaType);

— communication rate (CommRate);

— indicators (Indicators);

— MAC address (MACAddress);

— error log (ErrorLog).

6.1.2.3.3 UDP-IPObject

UDP-IPObject defines properties of UDP/IP used to implement ADS-net. UDP-IPObject attributes provide.

— IP address (IPADDR);

— multicast group information (UDP-IP-MCGs);

— host name (HostName);

— error log (ErrorLog).

6.1.2.4 Network management

6.1.2.4.1 General

The ADS-net NetworkManagement class represents the network configuration and performance adjustment capabilities of the ADS-net communication network integration model.

It is further divided into several classes, as shown in Figure 2.

6.1.2.4.2 Nm-Configuration

6.1.2.4.2.1 General

Nm-Configuration defines properties of network configuration related to ADS-net. Nm-Configuration attributes provide in particular:

— active data field number list (ActiveDataFieldNoList);

— active node number list (ActiveNodeNoList);

— active multicast group number list (ActiveMulticastGroupNoList).

6.1.2.4.2.2 Nm-MessageSelection

Nm-MessageSelection defines properties used for the management of message selection. Nm-MessageSelection attributes provide in particular:

— active producer TCD supported list (ActiveProducerTCDSupportedList);

— active consumer TCD supported list (ActiveConsumerTCDSupportedList).

6.1.2.4.2.3 Nm-Performance

Nm-Performance defines properties used for performance monitoring. Nm-Performance attributes provide:

— number of messages exchanged (MessageTransaction).

6.1.2.4.3 Nm-Fault

Nm-Fault defines properties used for fault monitoring. Nm-Fault attributes provide in particular:

— fault node list (FaultNodeList).

6.2 FL-net

6.2.1 Device profile

6.2.1.1 General

Figure 4 shows the class structure of the FL-net device profile.

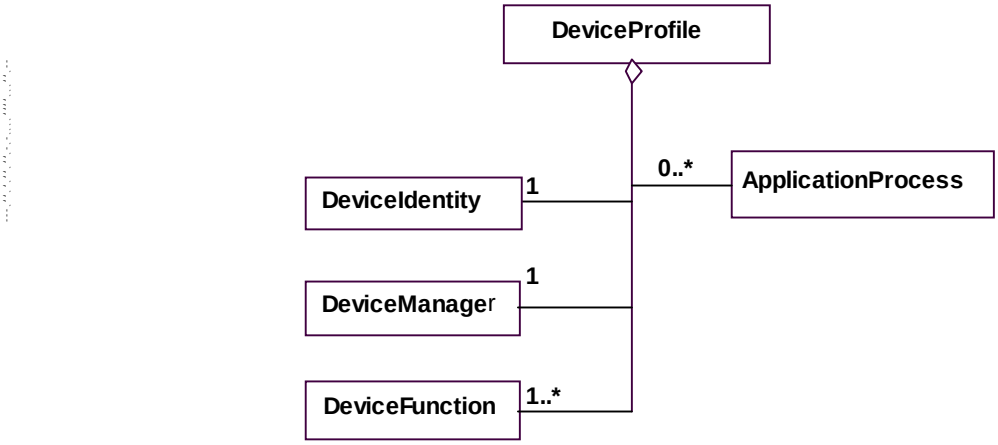

Figure 4 — FL-net device profile class diagram

The available formats for FL-net device profiles are described in B.1.

The XML schema representing the FL-net device profiles template is defined in B.1.5. The file name of this XML schema shall be “FL-net_Device_Profile.xsd”.

NOTE 1 The FL-net device profile class diagrams shown in Figure 4 define the main classes. Some classes are further decomposed; details are defined in Annex B.

NOTE 2 All these classes are mapped to the same XML schema defined in B.1.5.

6.2.1.2 Device identity

The DeviceIdentity class is defined in Figure 5, and the sub-classes are described in Table 2.

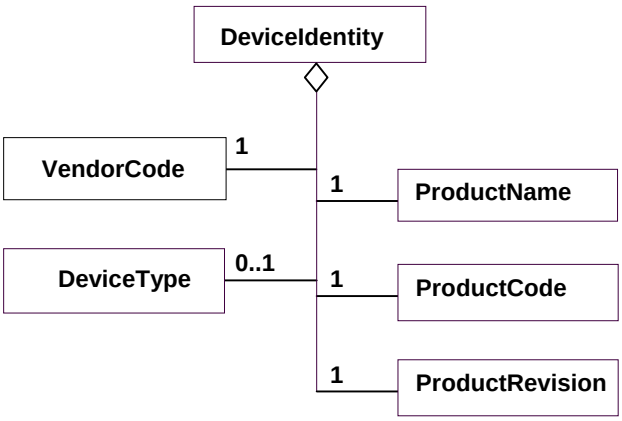

Figure 5 — FL-net DeviceIdentity class diagram

Table 2 — Elements of DeviceIdentity object

Class elements	Description
VendorCode	Code of the manufacturer or vendor of the device
ProductName	Vendor specific name of the product
ProductCode	Unique ID, identifying the product. The format is at the vendor's discretion
ProductRevision	Vendor specific revision number of the product
DeviceType	FL-net specific Device type (see Table B.1)

6.2.1.3 Device manager

6.2.1.3.1 General

Figure 6 shows the class structure of the DeviceManager object.

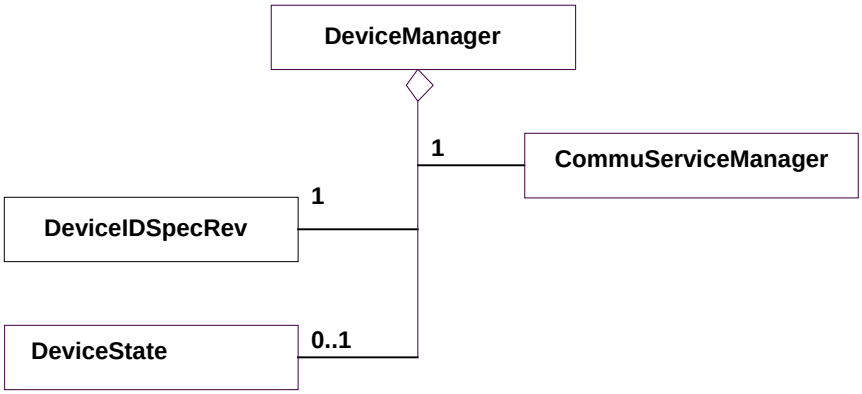

Figure 6 — FL-net DeviceManager class diagram

6.2.1.3.2 DeviceIDSpecRev

The DeviceIDSpecRev object shall describe the revision of FL-net identity object.

6.2.1.3.3 CommuServiceManager

The CommuServiceManager object shall describe the communication service, which is responsible for monitoring and configuration.

6.2.1.3.4 DeviceState

The DeviceState object shall describe the states of the device.

6.2.1.4 Device function

The DeviceFunction object contains attributes and supports services that enable the management (e.g. configuration) of a function of the device.

NOTE The device function class is not defined in ISO 15745-4.

6.2.1.5 Application process

The ApplicationProcess object contains attributes and supports services that correspond to the application requirement.

These attributes provide in particular:

— plant name (PlantName).

To allow specific representation of the application process, additional sub-classes and additional attributes may be defined to describe the application process.

6.2.2 Communication network profile

6.2.2.1 General

Figure 7 shows the class structure of the FL-net communication network profile.

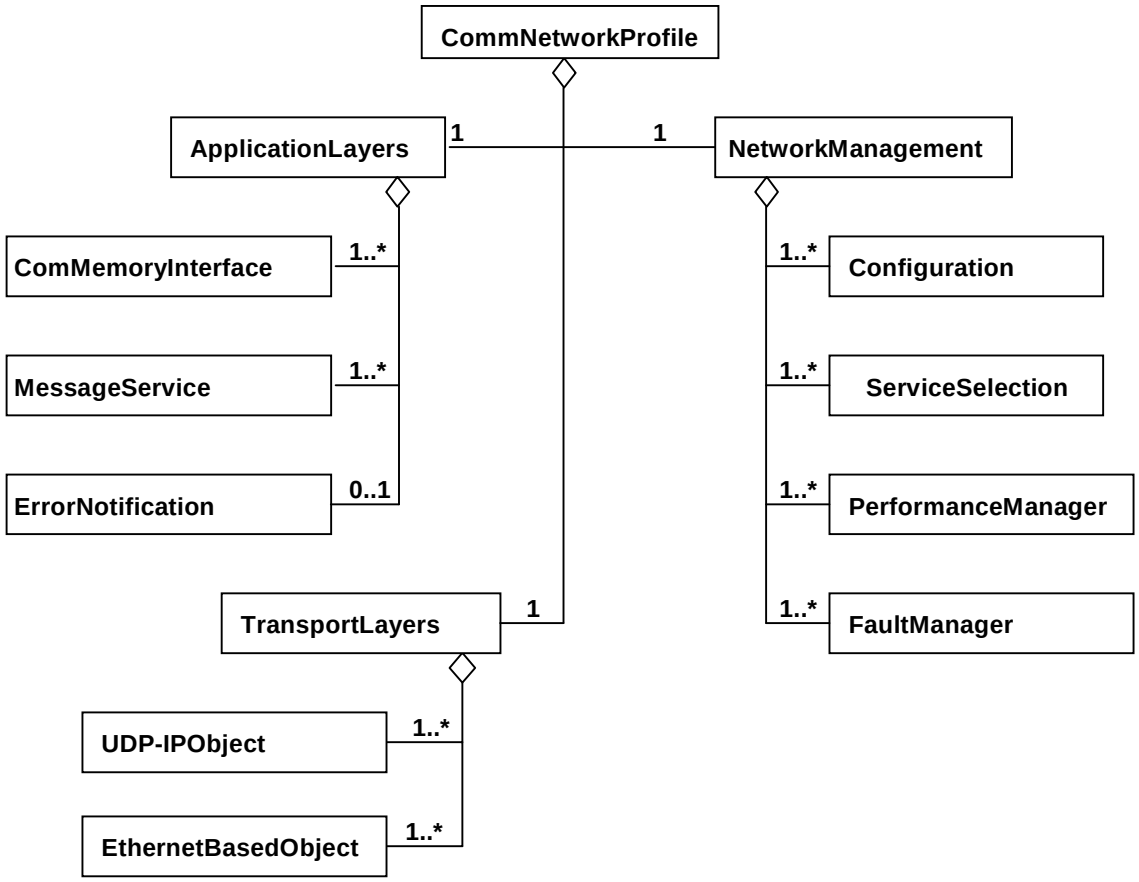

Figure 7 — FL-net communication network profile class diagram

The available formats for FL-net communication network profiles are described in B.2.

The XML schema representing the FL-net communication network profile template is defined in B.2.4.5. The file name of this XML schema shall be “FL-net_CommNet_Profile.xsd”.

6.2.2.2 ApplicationLayers

6.2.2.2.1 General

The FL-net ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the FL-net communication network integration model. It states the supported application service elements and their associated services.

It is further divided into several classes, as shown in Figure 7.

NOTE The ApplicationLayers object is fully specified in JEM 1479:2002.

6.2.2.2.2 ComMemoryInterface

The ComMemoryInterface object defines the properties associated with common memory interface. The elements of the ComMemoryInterface object are specified in B.2.2.1.

6.2.2.2.3 MessageService

The MessageService object defines the properties associated with message services in the device. The elements of the MessageService object are specified in B.2.2.2.

6.2.2.2.4 ErrorNotification

The ErrorNotification object defines the properties associated with error items related to the network and the device. The elements of the ErrorNotification object are specified in B.2.2.3.

6.2.2.3 TransportLayers

6.2.2.3.1 General

The FL-net TransportLayers class represents the combined profiles for the lower 4 OSI layers of the FL-net communication network integration model.

It is further divided into several classes, as shown in Figure 7.

6.2.2.3.2 EthernetBasedObject

The EthernetBasedObject object defines the properties associated with physical layer of FL-net. The elements of the EthernetBasedObject object are specified in B.2.3.1.

6.2.2.3.3 UDP-IPObject

The UDP-IPObject object defines the properties associated with data link layer configuration and monitoring. The elements of the UDP-IPObject object are specified in B.2.3.2.

6.2.2.4 NetworkManagement

6.2.2.4.1 General

The FL-net NetworkManagement class represents the network configuration and performance adjustment capabilities of the FL-net communication network integration model.

It is further divided into several classes, as shown in Figure 7.

NOTE The NetworkManagement object is fully specified in JEM 1479:2002.

6.2.2.4.2 Configuration

The Configuration object defines the properties associated with the initial set-up and modification of the configuration. The elements of the Configuration object are specified in B.2.4.1.

6.2.2.4.3 ServiceSelection

The ServiceSelection object defines the properties associated with network communication services. The elements of the ServiceSelection object are specified in B.2.4.2.

6.2.2.4.4 PerformanceManager

The PerformanceManager object defines the properties associated with performance of data exchange in the network. The elements of the PerfomanceManager object are specified in B.2.4.3.

6.2.2.4.5 FaultManager

The FaultManager object defines the properties associated with fault items of FL-net. The elements of the FaultManager object are specified in B.2.4.4.

6.3 EtherNet/IP

6.3.1 Device profile

6.3.1.1 General

Figure 8 shows the class structure of the EtherNet/IP device profile.

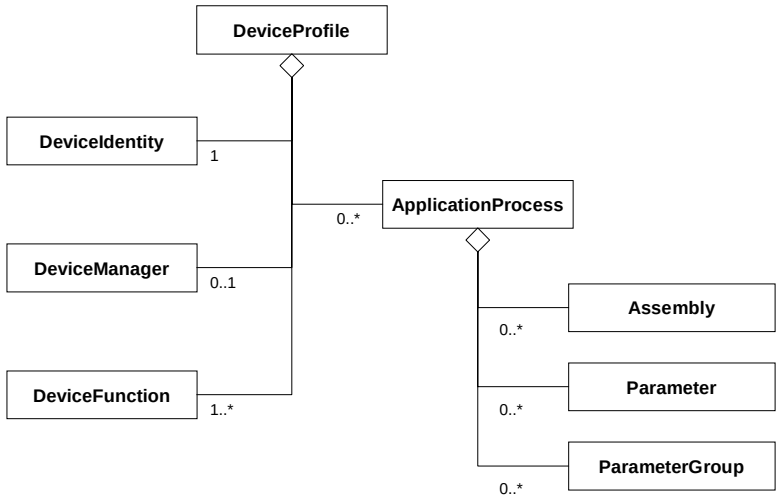

Figure 8 — EtherNet/IP device profile class diagram

The available formats for EtherNet/IP device profiles are described in C.2.

The XML schema representing the EtherNet/IP device profile template is defined in C.2.1.3.3. The file name of this XML schema shall be “CIP_Device_Profile.xsd”.

NOTE The EtherNet/IP device profile class diagram shown in Figure 8 defines the main classes. These classes are further decomposed; details are defined in Annex C.

The XML schema representing the encapsulation of a legacy EtherNet/IP EDS into the ISO 15745 device profile template is defined in C.2.2.2. The file name of this XML schema shall be “EDS_Device_Profile_wrapper.xsd”. The legacy EDS ASCII syntax itself is described in C.4.

6.3.1.2 Device identity

The DeviceIdentity class contains attributes which uniquely identify the device, and supports services which allow the retrieval of this information from the device.

These attributes provide in particular:

— manufacturer's identification (name and identification code);

— device identification (device type, product name, revision, serial number);

— device classification;

— location of storage of additional information (e.g. icons).

6.3.1.3 Device manager

The DeviceManager class contains attributes and supports services used to monitor and configure the device.

These attributes provide in particular:

— revision of the EtherNet/IP identity object;

— information on device structure (for devices integrated in a modular system).

Services allow:

— device reset;

— retrieval of DeviceManager attributes.

6.3.1.4 Device function

The DeviceFunction class contains attributes and supports services which enable the management (e.g. configuration) of a function of the device.

EXAMPLE Examples of DeviceFunction objects are Overload, Presence Sensing, Analogue Input, and Discrete Output objects.

NOTE The DeviceFunction class is not defined in ISO 15745-4.

6.3.1.5 Application process

Figure 9 shows the class structure of the ApplicationProcess class.

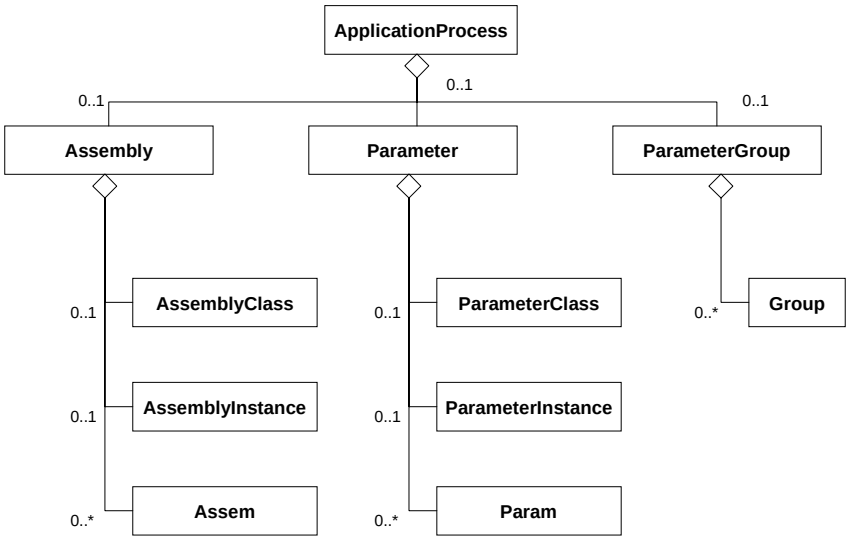

Figure 9 — EtherNet/IP ApplicationProcess class diagram

The Assembly class assembles several application process data items into a single block for optimisation of communications. The Parameter class provides a standardized interface for accessing individual application process data items. The ParameterGroup class specifies groups of related parameters for a specific purpose (e.g. configuration, monitoring).

The Assembly class and the Parameter class support attributes and services both at the class and instance levels.

The Assem, Param and Group classes specify individual instances of the main classes.

NOTE The Assembly class and the Parameter class correspond to the EtherNet/IP Assembly object and Parameter objects. The Assembly object is fully specified in IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2).

6.3.2 Communication network profile

6.3.2.1 General

Figure 10 shows the class structure of the EtherNet/IP communication network profile.

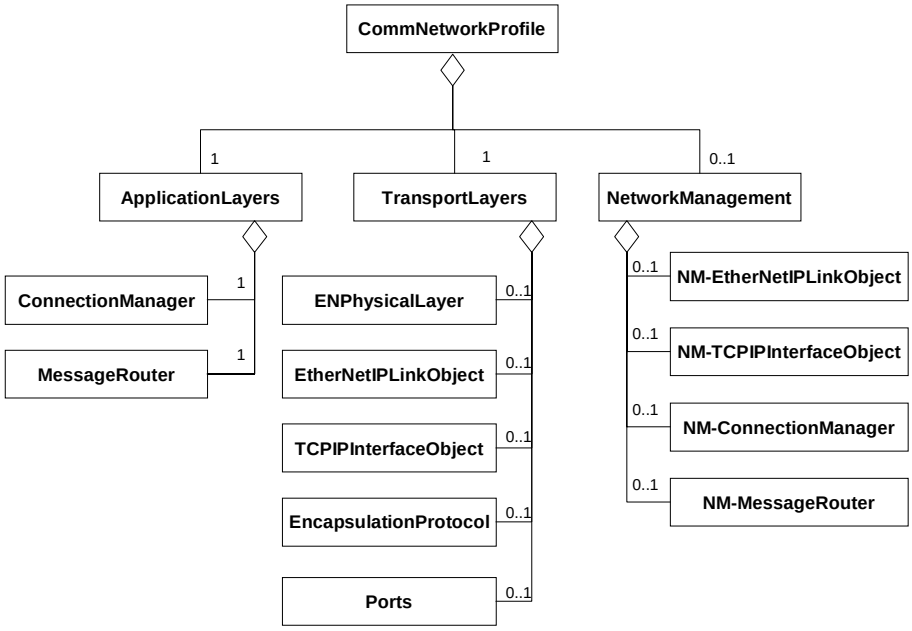

Figure 10 — EtherNet/IP communication network profile class diagram

The available formats for EtherNet/IP communication network profiles are described in C.3.

The XML schema representing the EtherNet/IP communication network profile template is defined in C.3.1.3. The file name of this XML schema shall be “ENet_CommNet_Profile.xsd”.

The XML schema representing the encapsulation of a legacy EtherNet/IP EDS into the ISO 15745 communication network profile template is defined in C.3.2.2. The file name of this XML schema shall be “EDS_CommNet_Profile_wrapper.xsd”. The legacy EDS ASCII syntax itself is described in C.4.

6.3.2.2 Application layers

The EtherNet/IP ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the EtherNet/IP communication network integration model.

It is further divided into several classes, as shown in Figure 10:

— ConnectionManager defines the properties associated with connections and connection management;

— MessageRouter defines the properties associated with internal message routing in the device.

NOTE The corresponding Connection Manager object and Message Router object are fully specified in IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2).

6.3.2.3 Transport layers

The EtherNet/IP TransportLayers class represents the combined profiles for the lower 4 OSI layers of the EtherNet/IP communication network integration model.

It is further divided into several classes, as shown in Figure 10:

— ENPhysicalLayer identifies the physical layer characteristics;

— EtherNetIPLinkObject defines the properties associated with data link layer configuration and monitoring;

— TCPIPInterfaceObject defines the properties associated with TCP/IP configuration and monitoring;

— EncapsulationProtocol defines the properties associated with encapsulation of application messages into TCP/IP;

— Ports identifies the device ports which are able to route messages from one link to another link.

NOTE The corresponding EtherNet Link object and TCP/IP Interface object are fully specified in IEC 61158-4:2003 (Type 2), the Encapsulation Protocol options are detailed in IEC 61158-6:2003 (Type 2).

6.3.2.4 Network management

The EtherNet/IP NetworkManagement class represents the network configuration and performance adjustment capabilities of the EtherNet/IP communication network integration model.

It is further divided into several classes, as shown in Figure 10:

— NM-EtherNetIPLinkObject, NM-TCPIPInterfaceObject, NM-ConnectionManager and NM-MessageRouter define the properties associated with class management of the corresponding objects.

Annex A
(normative)

ADS-net profile templates

A.1 General

The upper layers of the ADS-net are based on the Autonomous Decentralized System network (ADS-net). This protocol models all communication as messaging using producer-consumer relationships.

ADS-net provides access to all configuration data, status information, and runtime variables in a node and/or a system.

A.2 ADS-net device profile template description

A.2.1 General

The device profile XML files shall comply with the device profile XML schema as specified in A.2.3.

Contents of this XML schema are derived from the device profile class diagrams shown in 6.1.1, and extended with additional elements to allow full description of communication network requirements or capabilities.

A.2.2 Semantics of XML schema elements

A.2.2.1 DeviceIdentity

The semantics of the DeviceIdentity element are specified in Table A.1. These are used in the ADS-net run-time environment to provide information for full device identification.

Table A.1 — DeviceIdentity elements

Elements	Description and semantics
VendorName	Name of the device vendor
ProductCode	Identifier of the device in the vendor's product (maximum six characters)
ProductRevision	Revision number of the product
DeviceName	Name of the device

See Bibliography item [4] for more detail about the semantics of each attribute.

A.2.2.2 DeviceManager

The semantics of the DeviceManager-Attributes sub-elements of the DeviceManager element are specified in Table A.2. These are used in the ADS-net run-time environment.

Table A.2 — DeviceManager-Attributes elements

Elements	Description and semantics
DeviceState	Status of the device

A.2.2.3 DeviceFunction

The semantics of the DeviceFunction-Attributes sub-elements of the DeviceFunction element are specified in Table A.3. These are used in the ADS-net run-time environment.

Table A.3 — DeviceFunction-Attributes elements

Elements	Description and semantics
DFNO	Data field number (Decimal notation: 0 to 255)
LNODENO	Logical node number (Decimal notation: 0 to 4095)
MODE	Message mode (Decimal notation: 0 to 1) 0 = online mode 1 = test mode

See Bibliography item [4] for more detail about the semantics of each attribute.

A.2.2.4 ApplicationProcess

The semantics of the ApplicationProcess-Attributes sub-elements of the ApplicationProcess element are specified in Table A.4.

Table A.4 — ApplicationProcess-Attributes elements

Elements	Description and semantics
Producer-TCD-List	List of TCDs (Decimal notation: 1 to 65534) for message sent from the application process
Consumer-TCD-List	List of TCDs (Decimal notation: 1 to 65534) for message received by the application process
Priority	Priority (Decimal notation: 0 to 7) associated with a message, used both when sending and receiving the message

See Bibliography item [4] for more detail about the semantics of each attribute.

A.2.3 ADS-net_Device_Profile.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:annotation>
      <xsd:documentation>
         Device Profile - ISO 15745 Part 4
      </xsd:documentation>
   </xsd:annotation>

   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>

   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>

   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:annotation>
      <xsd:documentation>
         ADS-net Device Profile Template
      </xsd:documentation>
   </xsd:annotation>

   <xsd:element name="ProfileHeader">
```

```
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceIdentity" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="DeviceManager" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="DeviceFunction" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceIdentity">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="VendorName" type="xsd:string"/>
            <xsd:element name="ProductCode" type="xsd:string"/>
            <xsd:element name="ProductRevision" type="xsd:string"/>
            <xsd:element name="DeviceName" minOccurs="0" maxOccurs="1" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceManager">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceManager-Attributes" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="DeviceManager-Operations" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DeviceManager-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="DeviceState" minOccurs="1" maxOccurs="1" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceManager-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_Single" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Set_Attribute_Single" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceFunction">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceFunction-Attributes" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="DeviceFunction-Operations" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceFunction-Attributes">
      <xsd:complexType>
         <xsd:sequence>
```

```
            <xsd:element name="DFNO" minOccurs="1" maxOccurs="unbounded" type="xsd:unsignedByte"/>
            <xsd:element name="LNODENO" minOccurs="1" maxOccurs="unbounded" type="xsd:unsignedShort"/>
            <xsd:element name="MODE" minOccurs="1" maxOccurs="1" type="xsd:unsignedShort"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="DeviceFunction-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Get-Attribute-All" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Set-Attribute-All" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Create" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Delete" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="ApplicationProcess">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ApplicationProcess-Attributes" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="ApplicationProcess-Operations" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="ApplicationProcess-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Producer-TCD-List" minOccurs="1" maxOccurs="unbounded" type="xsd:string"/>
            <xsd:element name="Consumer-TCD-List" minOccurs="1" maxOccurs="unbounded" type="xsd:string"/>
            <xsd:element name="Priority" minOccurs="1" maxOccurs="unbounded" type="xsd:unsignedShort"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="ApplicationProcess-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Get-Attribute-All" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Set-Attribute-All" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Create" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Delete" minOccurs="0" maxOccurs="1"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

A.3 ADS-net communication network profile template description

A.3.1 General

The communication network profile XML files shall comply with the communication network profile XML schema as specified in A.3.3.

Contents of this XML schema are derived from the communication network profile class diagrams shown in 6.1.2, and extended with additional elements to allow full description of communication network requirements or capabilities.

A.3.2 Semantics of XML schema elements

A.3.2.1 ApplicationLayers

A.3.2.1.1 DataField

A.3.2.1.1.1 General

This element specifies the supported instance attributes used to control a data field.

DataField correspond to a domain where subsystems (i.e. node computers or application programs) share information by peer-to-peer message exchange. The messages are uniquely identified within a data field. Node computers can share information by specifying the data field number as part of the message identifier. One data field is implemented per network address or sub-network address for communication among node computers, or implemented in memory for communication among application programs in a node computer.

A data field is uniquely identified by its Data Field Number (DFNO). A unique DFNO is assigned to each of the data fields within a system, in the range of 1 to 255. DFNO 0 is reserved for communication within the current node.

The semantics of the DataField-Attributes sub-elements of the DataField element are specified in Table A.5.

Table A.5 — DataField-Attributes elements

Elements	Description and semantics
DFNO	Data field number (Decimal notation: 0 to 255)
NETADDR	Subnet address described by the Internet standard dot format like ” XXX.XXX.XXX.XXX”
BCADDR	Broadcast address described by the Internet standard dot format
NETMASK	Subnet mask described by the Internet standard dot format
MCG-Table	List of MCG numbers (Decimal notation: 0 to 255) used in this data field

See Bibliography item [4] for more detail about the semantics of each attribute.

A.3.2.1.1.2 AliveNotification

This element specifies the attributes used for health check of a node computer.

The semantics of the AliveNotification element are specified in Table A.6.

Table A.6 — AliveNotification elements

Elements	Description and semantics
ND-Name	Node name (ASCII character string up to nine characters)
OS-Name	Vendor device name (ASCII character string up to nine characters). Adherence to the following guideline is recommended: "BN_MN" BN: Identifier of the vendor (2 characters) MN: Vendor-defined device or OS name
TM-Out	Alive-signal timeout monitoring interval in seconds. The interval of time elapsed before a node is recognized as dead after it ceases to transmit an alive-signal.
Chg-Time	Time at which node state changes occurred. Use of the Greenwich Mean Time (elapsed time in seconds since 0 hour 0 minute 0 second, 1970) is recommended. Specify 0 if the Greenwich Mean Time cannot be used.
IPADDR	The IP address of LAN described by the Internet standard dot format

See Bibliography item [4] for more detail about the semantics of each attribute.

A.3.2.1.1.3 ErrorNotification

This element specifies the attributes used for fault check of a node computer.

The semantics of the ErrorNotification element are specified in Table A.7.

Table A.7 — ErrorNotification elements

Elements	Description and semantics
CNT-Mode-Alive	Number of modules reporting that they are alive
Mod-Alive-List	Status list for all modules (dead or alive)
CNT-Error	Number of reported errors
ERR-Name	Error name
ERR-List	List of error numbers
Mod-NO	Module number (1 or higher)
ERR-NO	Error number, which is uniquely identified by ERR-Name
CNT-Option	The length of optional information
Option-List	List of optional information

See Bibliography item [4] for more detail about the semantics of each attribute.

A.3.2.1.2 MessageSelection

This element specifies the attributes used for message exchange in ADS-net.

A message is uniquely identified by a Transaction Code (TCD). The transmitter sends a message associated with a TCD using a multicast transmission to a multicast group within a specified data field. The nodes within the specified multicast group accept only the messages with the relevant TCD.

The semantics of the MessageSelection-Attributes sub-elements of the MessageSelection element are specified in Table A.8.

Table A.8 — MessageSelection-Attributes elements

Elements	Description and semantics
Producer-TCD-List	List of TCDs (Decimal notation: 1 to 65534) for the messages to be sent
Consumer-TCD-List	List of TCDs (Decimal notation: 1 to 65534) for the messages to be received
Producer-MCG-List	List of MCG numbers (Decimal notation: 0 to 255) used to send messages
Consumer-MCG-List	List of MCG numbers (Decimal notation: 0 to 255) used to receive messages

See Bibliography item [4] for more detail about the semantics of each description.

A.3.2.2 TransportLayers

A.3.2.2.1 EthernetObject

This element specifies the attributes of Ethernet used for ADS-net.

The EthernetObject defines the attributes associated with data link layer configuration and monitoring.

The semantics of the EthernetObject-Attributes sub-elements of the EthernetObject element are specified in Table A.9.

Table A.9 — EthernetObject-Attributes elements

Elements	Description and semantics
MediaType	The type of Ethernet media (e.g. 10Base-T, 100Base-TX)
CommRate	Communication rate depending on the Ethernet media (e.g. 100Mbps)
Indicators	Status of the Ethernet media
MACAddress	MAC address
ErrorLog	List of errors that occurred in the physical and data link layers

A.3.2.2.2 UDP-IPObject

This element specifies the attributes of UDP/IP used for ADS-net.

The semantics of the UDP-IPObject-Attributes sub-elements of the UDP-IPObject element are specified in Table A.10.

Table A.10 — UDP-IPObject -Attributes elements

Elements	Description and semantics
IPADDR	IP address described by the Internet standard dot format
UDP-IP-MCGs	The semantics of this element are specified in Table A.11
HostName	Host name of the device corresponding to the IP address
ErrorLog	List of errors that occurred in the transport layer

Table A.11 — UDP-IP-MCGs elements

Elements	Description and semantics
McgNumber	The identification number of the multicast group (Decimal notation: 0 to 255)
PortNumber	Port number used for the multicast group

See Bibliography item [4] for more detail about the semantics of each attribute.

A.3.2.3 NetworkManagement

A.3.2.3.1 Nm-Configuration

This element specifies the attributes used for network configuration.

The semantics of the Nm-Configuration-Attributes sub-elements of the Nm-Configuration element are specified in Table A.12.

Table A.12 — Nm-Configuration -Attributes elements

Elements	Description and semantics
ActiveDataFieldNoList	List of data field numbers (Decimal notation: 0 to 255) used in the target system
ActiveNodeNoList	List of node numbers (Decimal notation: 0 to 4095) operating in the target system. This list is constructed by gathering alive message information.
ActiveMulticastGroupNoList	List of MCG numbers (Decimal notation: 0 to 255) used in the target system

See Bibliography item [4] for more detail about the semantics of each description.

A.3.2.3.2 Nm-MessageSelection

This element specifies the attributes used for message selection.

The semantics of the Nm-MessageSelection-Attributes sub-elements of the Nm-MessageSelection element are specified in Table A.13.

Table A.13 — Nm-MessageSelection-Attributes elements

Elements	Description and semantics
ActiveProducerTCDSupportedList	List of TCDs (Decimal notation: 1 to 65534) used for any messages to be sent
ActiveConsumerTCDSupportedList	List of TCDs (Decimal notation: 1 to 65534) used for any messages to be received
ActiveProducerMulticastGroupSupportedList	List of MCG numbers (Decimal notation: 0 to 255) used for any message transmission
ActiveConsumerMulticastGroupSupportedList	List of MCG numbers (Decimal notation: 0 to 255) used for any message reception

See Bibliography item [4] for more detail about the semantics of each description.

A.3.2.3.3 Nm-Performance

This element specifies the attributes used for performance monitoring.

The semantics of the Nm-Performance-Attributes sub-elements of the Nm-Performance element are specified in Table A.14.

Table A.14 — Nm-Performance-Attributes elements

Elements	Description and semantics
MessageTransaction	The number of messages sent and received by a device

A.3.2.3.4 Nm-Fault

This element specifies the attributes used for fault monitoring.

The semantics of the Nm-Fault-Attributes sub-elements of the Nm-Fault element are specified in Table A.15.

Table A.15 — Nm-Fault-Attributes elements

Elements	Description and semantics
FaultNodeList	List of the node numbers (Decimal notation: 0 to 4095), which are diagnosed as faulty
NotificationOfFaultProgram	Identifier of the faulty module detected by the alive message

See Bibliography item [4] for more detail about the semantics of each description.

A.3.3 ADS-net_CommNet_Profile.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:annotation>
      <xsd:documentation>
         Communication Profile - ISO 15745 Part 4
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
```

```
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>
         ADS-net Communication Network Profile Template
      </xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ApplicationLayers"/>
            <xsd:element ref="TransportLayers"/>
            <xsd:element ref="NetworkManagement" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ApplicationLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DataField" maxOccurs="unbounded"/>
            <xsd:element ref="MessageSelection" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DataField">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DataField-Attributes"/>
            <xsd:element ref="DataField-Operations" minOccurs="0"/>
            <xsd:element ref="AliveNotification"/>
            <xsd:element ref="ErrorNotification" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DataField-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="DFNO" type="xsd:unsignedByte"/>
            <xsd:element name="NETADDR" type="xsd:unsignedInt"/>
            <xsd:element name="BCADDR" type="xsd:unsignedInt"/>
```

```
            <xsd:element name="NETMASK" type="xsd:unsignedInt"/>
            <xsd:element name="MCG-Table" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DataField-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Set_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Set_Attribute_All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="AliveNotification">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ND-Name" type="xsd:string"/>
            <xsd:element name="OS-Name" type="xsd:string"/>
            <xsd:element name="TM-Out" type="xsd:nonNegativeInteger"/>
            <xsd:element name="Chg-Time" type="xsd:nonNegativeInteger"/>
            <xsd:element name="IPADDR" type="xsd:unsignedInt"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ErrorNotification">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="CNT-Mode-Alive" type="xsd:string"/>
            <xsd:element name="Mod-Alive-List" type="xsd:string" minOccurs="0"/>
            <xsd:element name="CNT-Error" type="xsd:string"/>
            <xsd:element name="ERR-Name" type="xsd:string" minOccurs="0"/>
            <xsd:element name="ERR-List" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Mod-NO" type="xsd:string" minOccurs="0"/>
            <xsd:element name="ERR-NO" type="xsd:string" minOccurs="0"/>
            <xsd:element name="CNT-Option" type="xsd:string"/>
            <xsd:element name="Option-List" type="xsd:string" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageSelection">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="MessageSelection-Attributes"/>
            <xsd:element ref="MessageSelection-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageSelection-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Producer-TCD-List" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Consumer-TCD-List" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Producer-MCG-List" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Consumer-MCG-List" type="xsd:string" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageSelection-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Set_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Set_Attribute_All" minOccurs="0"/>
            <xsd:element name="Create" minOccurs="0"/>
            <xsd:element name="Delete" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="TransportLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="EthernetObject" maxOccurs="unbounded"/>
            <xsd:element ref="UDP-IPObject" maxOccurs="unbounded"/>
```

```
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="EthernetObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="EthernetObject-Attributes"/>
         <xsd:element ref="EthernetObject-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="EthernetObject-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="MediaType" type="xsd:string"/>
         <xsd:element name="CommRate" type="xsd:unsignedInt"/>
         <xsd:element name="Indicators" type="xsd:string" minOccurs="0"/>
         <xsd:element name="MACAddress" type="xsd:string"/>
         <xsd:element name="ErrorLog" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="EthernetObject-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Get_Attribute_All" minOccurs="0"/>
         <xsd:element name="Set_Attribute_All" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="UDP-IPObject-Attributes"/>
         <xsd:element ref="UDP-IPObject-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="IPADDR" type="xsd:string"/>
         <xsd:element ref="UDP-IP-MCGs" maxOccurs="unbounded"/>
         <xsd:element name="HostName" type="xsd:string"/>
         <xsd:element name="ErrorLog" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IP-MCGs">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="McgNumber" type="xsd:unsignedShort"/>
         <xsd:element name="PortNumber" type="xsd:unsignedShort"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Get_Attribute_All" minOccurs="0"/>
         <xsd:element name="Set_Attribute_All" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="NetworkManagement">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="Nm-Configuration" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="Nm-MessageSelection" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="Nm-Performance" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="Nm-Fault" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
```

```
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Configuration">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="Nm-Configuration-Attributes"/>
            <xsd:element ref="Nm-Configuration-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Configuration-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ActiveDataFieldNoList" type="xsd:string"/>
            <xsd:element name="ActiveNodeNoList" type="xsd:string"/>
            <xsd:element name="ActiveMulticastGroupNoList" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Configuration-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Set_Attribute_All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-MessageSelection">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="Nm-MessageSelection-Attributes"/>
            <xsd:element ref="Nm-MessageSelection-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-MessageSelection-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ActiveProducerTCDSupportedList" type="xsd:string"/>
            <xsd:element name="ActiveConsumerTCDSupportedList" type="xsd:string"/>
            <xsd:element name="ActiveProducerMulticastGroupSupportedList" type="xsd:string"/>
            <xsd:element name="ActiveConsumerMulticastGroupSupportedList" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-MessageSelection-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Set_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Set_Attribute_All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Performance">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="Nm-Performance-Attributes" minOccurs="0"/>
            <xsd:element ref="Nm-Performance-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Performance-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="MessageTransaction" type="xsd:unsignedInt"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Performance-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
```

```
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Fault">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="Nm-Fault-Attributes" minOccurs="0"/>
            <xsd:element ref="Nm-Fault-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Fault-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="FaultNodeList" type="xsd:string"/>
            <xsd:element name="NotificationOfFaultProgram" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Nm-Fault-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get-Attributes-All" minOccurs="0"/>
            <xsd:element name="Set-Attributes-All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Annex B
(normative)

FL-net profile templates

B.1 Device profile template description

B.1.1 General

The device profile template XML schema defined in B.1.5 contains the mapping of the device profile class diagrams shown in 6.2.1. Besides the mapped classes and attributes, it contains additional elements to enable full description of device requirements or capabilities.

B.1.2 DeviceIdentity

This element specifies the attributes and operations of the DeviceIdentity object, together with additional information for full device identification.

Table B.1 describes the elements of the DeviceType object.

Table B.1 — Elements of DeviceType object

Elements	Description and semantics
PC	Programmable controller
NC	Numerical control unit
RC	Robot controller
computer	Personal computer, panel computer, work station
SP-*xxx*	xxx is the vendor defined device name

B.1.3 DeviceManager

B.1.3.1 General

This element specifies the attributes and operations of the DeviceManager object, together with additional information for device management.

B.1.3.2 DeviceIDSpecRev object

Table B.2 describes the elements of the DeviceIdSpecRev object as defined in 6.2.1.3.2.

Table B.2 — Elements of DeviceIDSpecRev object

Elements	Description and semantics
COMVER	Revision of the DeviceIdentity
Date	Date of identity defined

B.1.3.3 CommuServiceManager object

Table B.3 describes the elements of the CommuServiceManager object as defined in 6.2.1.3.3.

Table B.3 — Elements of CommuServiceManager object

Elements	Description and semantics
DM-NodeNumber	Node number used in FL-net
DM-ComMemory1Datafield	Data field of the common memory area 1 used in the device
DM-ComMemory2Datafield	Data field of the common memory area 2 used in the device
DM-MessageServiceList	List of message services used in the device

B.1.3.4 DeviceState object

Table B.4 describes the elements of the DeviceState object as defined in 6.2.1.3.4.

Table B.4 — Elements of DeviceState object

Elements	Description and semantics
RUN-STOP	Indication of the device state (running or stopped)
Normal	Indication of the normal state of the device
Warning	Indication of the fault state of the device – device is still running
Alarm	Indication of the fault state of the device – device is not running
InRingState	Indication of the communication state of the device

B.1.4 ApplicationProcess

This element specifies the attributes and operations of the ApplicationProcess object, together with additional elements.

Table B.5 describes the elements of the PlantName object as defined in 6.2.1.5.

Table B.5 — Elements of PlantName object

Elements	Description and semantics
PlantName	Name of the plant or process

B.1.5 FL-net device profile template XML schemas

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:annotation>
      <xsd:documentation>
         Device Profile - ISO 15745 Part 4
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
```

```
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>
         FL-net Device Profile Template
      </xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceIdentity" minOccurs="0"/>
            <xsd:element ref="DeviceManager" minOccurs="0"/>
            <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
            <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DeviceIdentity">
```

```
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="DeviceIdentity-Attributes"/>
         <xsd:element ref="DeviceIdentity-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceIdentity-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="VendorCode" type="xsd:string"/>
         <xsd:element name="ProductName" type="xsd:string"/>
         <xsd:element name="ProductCode" type="xsd:string"/>
         <xsd:element name="ProductRevision" type="xsd:string"/>
         <xsd:element name="DeviceType" minOccurs="0">
            <xsd:simpleType>
               <xsd:union>
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="PC"/>
                        <xsd:enumeration value="NC"/>
                        <xsd:enumeration value="RC"/>
                        <xsd:enumeration value="computer"/>
                     </xsd:restriction>
                  </xsd:simpleType>
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:string">
                        <xsd:pattern value="SP-\*.{0,32}\*"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:union>
            </xsd:simpleType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceIdentity-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get_Attribute_All" minOccurs="0"/>
         <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceManager">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="DeviceManager-Attributes"/>
         <xsd:element ref="DeviceManager-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceManager-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="DeviceIDSpecRev">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="COMVER">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="1"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:element>
                  <xsd:element name="Date" type="xsd:date"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="CommuServiceManager">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="DM-NodeNumber">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="1"/>
```

```
                        <xsd:maxInclusive value="255"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:element>
               <xsd:element name="DM-ComMemory1Datafield">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:integer">
                        <xsd:minInclusive value="0"/>
                        <xsd:maxInclusive value="63"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:element>
               <xsd:element name="DM-ComMemory2Datafield">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:integer">
                        <xsd:minInclusive value="0"/>
                        <xsd:maxInclusive value="8195"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:element>
               <xsd:element name="DM-MessageServiceList" type="xsd:unsignedShort"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="DeviceState" minOccurs="0">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="RUN-STOP" type="xsd:boolean"/>
               <xsd:element name="Normal" type="xsd:boolean"/>
               <xsd:element name="Warning" type="xsd:boolean"/>
               <xsd:element name="Alarm" type="xsd:boolean"/>
               <xsd:element name="InRingState" type="xsd:boolean"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
   </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="DeviceManager-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get_Attribute_All" minOccurs="0"/>
         <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceFunction">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:any namespace="##any"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="ApplicationProcess">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="ApplicationProcess-Attributes"/>
         <xsd:element ref="ApplicationProcess-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="ApplicationProcess-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="PlantName" type="xsd:string"/>
         <xsd:choice maxOccurs="unbounded">
            <xsd:element ref="externalelement" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:choice>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="externalelement">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:any namespace="##any"/>
      </xsd:sequence>
```

```
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ApplicationProcess-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get-Attribute-All" minOccurs="0"/>
            <xsd:element name="Set-Attribute-All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

B.2 FL-net communication network profile template description

B.2.1 General

The communication network profile template XML schema defined in B.2.4.5 contains the mapping of the communication network profile class diagrams shown in Figure 7. Besides the mapped classes and attributes, it contains additional elements, to enable full description of communication network requirement or capabilities.

B.2.2 ApplicationLayers

B.2.2.1 ComMemoryInterface object

Table B.6 describes the elements of the ComMemoryInterface object as defined in 6.2.2.2.2. The ComMemory1AllocationList and ComMemory2AllocationList sub-elements provide memory allocation information. Their attributes are described in Table B.7.

Table B.6 — Elements of ComMemoryInterface

Elements	Description and semantics
ComMemory1AllocationList	The allocation of the common memory area 1
ComMemory2AllocationList	The allocation of the common memory area 2
NodeNumber	Node number
DataAvailability	Information of the data availability on the common memory
DataConsistency	Consistency of the data
Cyclic	Cyclical data exchange

Table B.7 — Attributes of ComMemoryAllocationList

Elements	Description and semantics
Size	Data size
FieldNumber	Common memory data field number

B.2.2.2 MessageService object

Table B.8 describes the elements of the MessageService object as defined in 6.2.2.2.3.

Table B.8 — Elements of MessageService object

Elements	Description and semantics
NodeNumberList	List of node numbers
TCDList	List of TCDs

B.2.2.3 ErrorNotification object

Table B.9 describes the elements of the ErrorNotification object as defined in 6.2.2.2.4.

Table B.9 — Elements of ErrorNotification object

Elements	Description and semantics
ActiveNodeNumberList	List of the active node numbers
NodeStatus	Status of node

B.2.3 TransportLayers

B.2.3.1 EthernetBasedObject object

Table B.10 describes the elements of the EthernetBasedObject object as defined in 6.2.2.3.2.

Table B.10 — Elements of EthernetBasedObject object

Elements	Description and semantics
MediaType	The type of Ethernet media (e.g. 10Base-5, 10Base-T, 100Base-TX)
CommRate	Communication rate depending on the Ethernet media (e.g. 10Mbps, 100Mbps)
Indicators	Status of the Ethernet media
MACAddress	MAC address
ErrorLog	List of errors that occurred in the physical and data link layers

B.2.3.2 UDP-IPObject object

Table B.11 describes the elements of the UDP-IPObject object as defined in 6.2.2.3.3.

Table B.11 — Elements of UDP-IPObject object

Elements	Description and semantics
IPADDR	IP address
PortNumber	Port number defined in FL-net
HostName	Host name of the device corresponding to the IP address
ErrorLog	List of errors that occurred in the transport layer

B.2.4 NetworkManagement

B.2.4.1 Configuration object

Table B.12 describes the elements of the Configuration object as defined in 6.2.2.4.2.

Table B.12 — Elements of Configuration object

Elements	Description and semantics
SpecificationMajorVersion	Major version of the FL-net specification
SpecificationMinorVersion	Minor version of the FL-net specification
InRingManager	Manager object associated with network participation
ConnectionManager	Manager object associated with connection
Indicators	Status of the FL-net

B.2.4.2 ServiceSelection object

Table B.13 describes the elements of the ServiceSelection object as defined in 6.2.2.4.3.

Table B.13 — Elements of ServiceSelection object

Elements	Description and semantics
NM-MessageTypeList	List of message types
NM-TCDList	List of TCDs for the messages to be sent
NM-VendorCodeList	List of vendor codes

B.2.4.3 PerformanceManager object

Table B.14 describes the elements of the PerformanceManager object as defined in 6.2.2.4.4.

Table B.14 — Elements of PerformanceManager object

Elements	Description and semantics
TokenMonitoringTime	Maximum time before the next node recreates a token
TokenHoldTime	Maximum time that a node is allowed to hold a token
MinimumFrameInterval	Minimum interval time between the transmission of frames
RefreshCycleAvailableTime	Refresh cycle time
MessageTransmissionSchedule	Schedule for message transmission

B.2.4.4 FaultManager object

Table B.15 describes the elements of the FaultManager object as defined in 6.2.2.4.5.

Table B.15 — Elements of FaultManager object

Elements	Description and semantics
FaultNodeList	List of the active node numbers
ErrorName	Error name, which is assigned to each error
ErrorLog	List of errors that occurred in the FL-net communication
ComMemoryAreaOverlap	Information on overlap of common memory allocation
NodeNumberConflict	Information on node number conflicts

B.2.4.5 XML schema : FL-net_CommNet_Profile.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:annotation>
      <xsd:documentation>
         Communication Profile - ISO 15745 Part 4
      </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>
         FL-net Communication Network Profile Template
      </xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
```

```
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ApplicationLayers"/>
            <xsd:element ref="TransportLayers"/>
            <xsd:element ref="NetworkManagement" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ApplicationLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ComMemoryInterface" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="MessageService" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="ErrorNotification" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ComMemoryInterface">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ComMemoryInterface-Attributes"/>
            <xsd:element ref="ComMemoryInterface-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ComMemoryInterface-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ComMemory1AllocationList" minOccurs="0" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:attribute name="Size">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="0"/>
                           <xsd:maxInclusive value="64"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:attribute>
                  <xsd:attribute name="FieldNumber">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="0"/>
                           <xsd:maxInclusive value="63"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:attribute>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ComMemory2AllocationList" minOccurs="0" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:attribute name="Size">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="0"/>
                           <xsd:maxInclusive value="8196"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:attribute>
                  <xsd:attribute name="FieldNumber">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                           <xsd:minInclusive value="0"/>
                           <xsd:maxInclusive value="8195"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:attribute>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="NodeNumber" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xsd:element name="DataAvailability" type="xsd:boolean" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="DataConsistency" type="xsd:boolean" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Cyclic" type="xsd:boolean" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ComMemoryInterface-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Set_Attribute_Single" minOccurs="0"/>
            <xsd:element name="Apply_Attributes" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageService">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="MessageService-Attributes"/>
            <xsd:element ref="MessageService-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageService-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="NodeNumberList" type="xsd:unsignedShort" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="TCDList" type="xsd:unsignedInt" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="MessageService-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ErrorNotification">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ActiveNodeNumberList" type="xsd:unsignedInt" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="NodeStatus" type="xsd:unsignedInt" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="TransportLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="EthernetBasedObject" maxOccurs="unbounded"/>
            <xsd:element ref="UDP-IPObject" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="EthernetBasedObject">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="EthernetBasedObject-Attributes"/>
            <xsd:element ref="EthernetBasedObject-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="EthernetBasedObject-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="MediaType" type="xsd:string"/>
            <xsd:element name="CommRate" type="xsd:unsignedInt"/>
            <xsd:element ref="Indicators" type="xsd:string" minOccurs="0"/>
            <xsd:element name="MACAddress" type="xsd:string"/>
            <xsd:element name="ErrorLog" type="xsd:unsignedInt" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
```

```
<xsd:element name="EthernetBasedObject-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Get-Attributes-All" minOccurs="0"/>
         <xsd:element name="Clear" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="UDP-IPObject-Attributes"/>
         <xsd:element ref="UDP-IPObject-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="IPADDR" type="xsd:string"/>
         <xsd:element name="PortNumber" type="xsd:unsignedShort" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element name="HostName" type="xsd:string"/>
         <xsd:element name="ErrorLog" type="xsd:unsignedInt" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="UDP-IPObject-Operations">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
         <xsd:element name="Get-Attributes-All" minOccurs="0"/>
         <xsd:element name="Clear" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="NetworkManagement">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="Configuration" maxOccurs="unbounded"/>
         <xsd:element ref="ServiceSelection" maxOccurs="unbounded"/>
         <xsd:element ref="PerformanceManager" maxOccurs="unbounded"/>
         <xsd:element ref="FaultManager" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="Configuration">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="Configuration-Attributes"/>
         <xsd:element ref="Configuration-Operations" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="Configuration-Attributes">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="SpecificationMajorVersion" minOccurs="0"/>
         <xsd:element ref="SpecificationMinorVersion" minOccurs="0"/>
         <xsd:element ref="InRingManager" minOccurs="0"/>
         <xsd:element ref="ConnectionManager" minOccurs="0"/>
         <xsd:element ref="Indicators" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="SpecificationMajorVersion">
   <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
         <xsd:minInclusive value="1"/>
         <xsd:maxInclusive value="15"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
<xsd:element name="SpecificationMinorVersion">
```

```
      <xsd:simpleType>
         <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="15"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:element>
   <xsd:element name="InRingState" type="xsd:string"/>
   <xsd:element name="ConnectionManager" type="xsd:string"/>
   <xsd:element name="Indicators" type="xsd:string"/>
   <xsd:element name="Configuration-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get-Attributes-All" minOccurs="0"/>
            <xsd:element name="Set-Attributes-All" minOccurs="0"/>
            <xsd:element name="Clear" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ServiceSelection">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ServiceSelection-Attributes"/>
            <xsd:element ref="ServiceSelection-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ServiceSelection-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="NM-MessageTypeList" type="xsd:unsignedShort" maxOccurs="unbounded"/>
            <xsd:element name="NM-TCDList" type="xsd:unsignedInt" maxOccurs="unbounded"/>
            <xsd:element name="NM-VendorCodeList" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ServiceSelection-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get_Attribute_All" minOccurs="0"/>
            <xsd:element name="Get_Attribute_Single" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="PerformanceManager">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="PerformanceManager-Attributes"/>
            <xsd:element ref="PerformanceManager-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="PerformanceManager-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="TokenMonitoringTime" type="xsd:unsignedShort" minOccurs="0"/>
            <xsd:element name="TokenHoldTime" type="xsd:unsignedShort" minOccurs="0"/>
            <xsd:element name="MinimumFrameInterval" minOccurs="0"/>
            <xsd:element name="RefreshCycleAvailableTime" type="xsd:unsignedShort" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="MessageTransmissionSchedule" type="xsd:unsignedShort" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="PerformanceManager-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get-Attributes-All" minOccurs="0"/>
            <xsd:element name="Set-Attributes-All" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
```

```
   </xsd:element>
   <xsd:element name="FaultManager">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="FaultManager-Attributes"/>
            <xsd:element ref="FaultManager-Operations" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="FaultManager-Attributes">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="FaultNodeList" type="xsd:unsignedShort" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="ErrorName" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ErrorLog" type="xsd:unsignedInt" minOccurs="0"/>
            <xsd:element name="ComMemoryAreaOverlap" type="xsd:boolean" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="NodeNumberConflict" type="xsd:boolean" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="FaultManager-Operations">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="Get-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Set-Attribute-Single" minOccurs="0"/>
            <xsd:element name="Get-Attributes-All" minOccurs="0"/>
            <xsd:element name="Set-Attributes-All" minOccurs="0"/>
            <xsd:element name="Clear" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Annex C
(normative)

EtherNet/IP profile templates

C.1 General

The upper layers of the EtherNet/IP network are based on the Common Industrial Protocol (CIP). This protocol models all communication and application entities as objects. CIP specific messaging requests services to be performed on corresponding object instances (or their attributes). This scheme provides an explicit access to all configuration, status, and runtime variables data in a node. At the same time, I/O connections allow direct exchange with the I/O database, without intermediate processing. In both cases, all data references within a device are specified using a CIP path, i.e. an octet string stream that defines the application object instance, attribute and/or connection end-point.

Multiple options are available for remote configuration of devices with a CIP communication interface, including:

— device information saved in printed or electronic format;

— dedicated Parameter Objects, which provide a known public interface to individual configuration/parameter data values, and may also embed additional configuration information such as descriptive text, data type, data limits and default;

— dedicated Configuration Assembly, which allows bulk upload and download of configuration data by grouping individual configuration/parameter data values;

— combinations of the above methods.

Configuration tools currently available for CIP-based devices use a specially formatted ASCII file, referred to as the Electronic Data Sheet (EDS), which provides:

— information needed to identify the connected device;

— a description of device data that can be accessed via the network (e.g. configurable parameters);

— a description of the communication capabilities supported by the device (e.g. connections);

— additional vendor-specific information.

The EDS allows a configuration tool to automate the device configuration process. The EDS requirements provide an open, consistent and compatible approach for performing device configuration in the CIP environment.

The EDS information is very similar to the information required in both communication network and device profiles, hence the following subclauses specify format for:

— communication network and device profile templates, as defined in ISO 15745-1;

— encapsulation of legacy EDS files in the ISO 15745 templates (“wrappers”);

— the legacy Electronic Data Sheet, including common semantics information.

NOTE The EtherNet/IP EDS (Electronic Data Sheet) of a given device can be derived from the contents of the corresponding XML device and communication network profile files, using the appropriate style sheets.

C.2 Device profile template description

C.2.1 Device profile template description – XML based

C.2.1.1 General

The device profile XML files shall comply with the device profile XML schema as specified in C.2.1.3.3.

Contents of this XML schema are derived from the device profile class diagrams shown in 6.3.1, and extended with additional elements to allow full description of device requirements or capabilities.

C.2.1.2 Semantics of XML schema elements

C.2.1.2.1 ProfileBody

This main element is associated with a set of attributes which provide additional information about the profile file.

The semantics of these attributes are specified in C.4.1.4.2.

C.2.1.2.2 DeviceIdentity

This element specifies the supported instance attributes and operations of the Identity Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)), together with additional information for full device identification. When appropriate, it also indicates the actual values of the instance attributes.

The semantics of the DeviceIdentity_InstanceAttributes sub-elements of the DeviceIdentity element are specified in Table C.1.

Table C.1 — DeviceIdentity_InstanceAttributes elements

XML schema elements	Object attributes	Semantics
SpecificationConformance	No	String specifying the reference version of the EtherNet/IP specifications
VendCode, ProdType, ProdCode, ProdRevision	Yes	See C.4.1.4.3
VendName, ProdTypeStr, ProdName, Catalog, Icon, ExcludeFromAdapterRackConnection	No	See C.4.1.4.3
Status, SerialNumber	Yes	Not applicable
State, ConfigurationConsistencyValue, HeartbeatInterval	Yes	Not applicable
DeviceClassification	No	See C.4.1.4.4 and C.4.2.2.1

C.2.1.2.3 DeviceManager

This element specifies the supported class attributes and operations of the Identity Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)), together with additional information for device management. When appropriate, it also indicates the actual values of the instance attributes.

The semantics of the Modular sub-element of the DeviceManager element are specified in C.4.1.5.2.

C.2.1.2.4 DeviceFunction

The contents of this element are not detailed in this document.

C.2.1.2.5 ApplicationProcess

C.2.1.2.5.1 Assembly

This element specifies the supported class and instance attributes and operations of the Assembly Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)), together with a description of the individual instances.

The semantics of the Assem, ProxyAssem and ProxiedAssem sub-elements of the Assembly element are specified in C.4.1.4.8 and C.4.1.5.3.2.

C.2.1.2.5.2 Parameter

This element specifies the supported class and instance attributes and operations of the Parameter Object, together with a description of the individual instances.

The semantics of the Parameter_ClassAttributes sub-element of the Parameter element are specified in C.4.1.4.5.

The semantics of the Param, ProxyParam and ProxiedParam sub-elements of the Parameter element are specified in C.4.1.4.6 and C.4.1.5.3.1.

C.2.1.2.5.3 ParameterGroup

This element specifies groups of related parameters for a specific purpose.

The semantics of the Group sub-element of the ParameterGroup element is specified in C.4.1.4.7.

C.2.1.3 XML schemas

C.2.1.3.1 MasterTemplateTypes.xsd

NOTE This XML schema contains all the styles defined as part of the master template in ISO 15745-1:2003.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!-- Target namespaces are not specified in this master template -->

<xsd:annotation>
   <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:simpleType name="ProfileClassID_DataType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AIP" />
      <xsd:enumeration value="Process" />
      <xsd:enumeration value="InformationExchange" />
      <xsd:enumeration value="Resource" />
      <xsd:enumeration value="Device" />
      <xsd:enumeration value="CommunicationNetwork" />
      <xsd:enumeration value="Equipment" />
      <xsd:enumeration value="Human" />
      <xsd:enumeration value="Material" />
   </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ISO15745Reference_DataType">
   <xsd:sequence>
      <xsd:element name="ISO15745Part" type="xsd:positiveInteger" />
      <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" />
      <xsd:element name="ProfileTechnology" type="xsd:string" />
   </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="IASInterface_DataType">
   <xsd:union>
```

```
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:enumeration value="CSI" />
            <xsd:enumeration value="HCI" />
            <xsd:enumeration value="ISI" />
            <xsd:enumeration value="API" />
            <xsd:enumeration value="CMI" />
            <xsd:enumeration value="ESI" />
            <xsd:enumeration value="FSI" />
            <xsd:enumeration value="MTI" />
            <xsd:enumeration value="SEI" />
            <xsd:enumeration value="USI" />
         </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
         <xsd:restriction base="xsd:string">
            <xsd:length value="4" />
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:union>
</xsd:simpleType>

<xsd:annotation>
   <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
</xsd:annotation>

<xsd:complexType name="ProfileHandle_DataType">
   <xsd:sequence>
      <xsd:element name="ProfileIdentification" type="xsd:string" />
      <xsd:element name="ProfileRevision" type="xsd:string" />
      <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1" />
   </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

C.2.1.3.2 CIPDataTypes.xsd

NOTE This XML schema defines the XML schema items (e.g. data types, element types, attribute groups) used in the other XML schemas.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <!-- Target namespaces are not specified in this master template -->
   <xsd:annotation>
      <xsd:documentation>* CIP DATA TYPES *</xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="dt_USINT">
      <xsd:restriction base="xsd:unsignedByte"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_UINT">
      <xsd:restriction base="xsd:unsignedShort"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_UDINT">
      <xsd:restriction base="xsd:unsignedInt"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_ULINT">
      <xsd:restriction base="xsd:unsignedLong"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_SINT">
      <xsd:restriction base="xsd:byte"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_INT">
      <xsd:restriction base="xsd:short"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_DINT">
      <xsd:restriction base="xsd:int"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_LINT">
      <xsd:restriction base="xsd:long"/>
   </xsd:simpleType>
   <xsd:simpleType name="dt_BYTE">
      <xsd:restriction base="xsd:hexBinary">
```

```
      <xsd:maxLength value="1"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="dt_WORD">
   <xsd:restriction base="xsd:hexBinary">
      <xsd:maxLength value="2"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="dt_DWORD">
   <xsd:restriction base="xsd:hexBinary">
      <xsd:maxLength value="4"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="dt_LWORD">
   <xsd:restriction base="xsd:hexBinary">
      <xsd:maxLength value="8"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="dt_REAL">
   <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="dt_LREAL">
   <xsd:restriction base="xsd:double"/>
</xsd:simpleType>
<xsd:simpleType name="dt_EDS_Char_Array">
   <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="dt_EPATH">
   <xsd:list itemType="et_EPATH_item"/>
</xsd:simpleType>
<xsd:simpleType name="dt_STRINGI">
   <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="dt_EDS_Date">
   <xsd:restriction base="xsd:date"/>
</xsd:simpleType>
<xsd:simpleType name="dt_EDS_Time_Of_Day">
   <xsd:restriction base="xsd:time"/>
</xsd:simpleType>
<xsd:simpleType name="dt_EDS_Revision">
   <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9]\.[1-9]|[1-9]\.[0-9]|[1-9]\.[1-9]"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="dt_EDS_URL">
   <xsd:restriction base="xsd:anyURI">
      <xsd:pattern value="http://.*"/>
      <xsd:pattern value="ftp://.*"/>
      <xsd:pattern value=".*"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="at_AccessType_OptionalGet">
   <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="None"/>
      <xsd:enumeration value="Get"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="at_AccessType_OptionalSet">
   <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="None"/>
      <xsd:enumeration value="Get"/>
      <xsd:enumeration value="Set"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="at_AccessType_Mandatory">
   <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Get"/>
      <xsd:enumeration value="Set"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="et_VendorSpecificKeyword">
   <xsd:restriction base="xsd:string">
      <xsd:pattern value="[1-9][0-9]{0,4}_([A-Z]|[a-z]|[0-9])([A-Z]|[a-z]|[0-9]|[_])*"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="et_EPATH_item">
```

```
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:pattern value="([0-9]|[a-f]|[A-F]){2}"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="SLOT"/>
               <xsd:enumeration value="SLOT_MINUS_ONE"/>
               <xsd:enumeration value="SYMBOL_ANSI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:pattern value="Param[1-9][0-9]{0,4}"/>
               <xsd:pattern value="\[Param[1-9][0-9]{0,4}\]"/>
               <xsd:pattern value="ProxyParam[1-9][0-9]{0,4}"/>
               <xsd:pattern value="\[ProxyParam[1-9][0-9]{0,4}\]"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:simpleType name="et_ParamReference">
      <xsd:restriction base="xsd:NMTOKEN">
         <xsd:pattern value="Param[1-9][0-9]{0,4}([:][0-9]{1,2})*"/>
         <xsd:pattern value="ProxyParam[1-9][0-9]{0,4}([:][0-9]{1,2})*"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:simpleType name="et_AssemReference">
      <xsd:restriction base="xsd:NMTOKEN">
         <xsd:pattern value="Assem[1-9][0-9]{0,4}"/>
         <xsd:pattern value="ProxyAssem[1-9][0-9]{0,4}"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:attributeGroup name="ag_FileDescription">
      <xsd:attribute name="DescText" type="dt_EDS_Char_Array" use="required"/>
      <xsd:attribute name="CreateDate" type="dt_EDS_Date" use="required"/>
      <xsd:attribute name="CreateTime" type="dt_EDS_Time_Of_Day" use="required"/>
      <xsd:attribute name="ModDate" type="dt_EDS_Date" use="optional"/>
      <xsd:attribute name="ModTime" type="dt_EDS_Time_Of_Day" use="optional"/>
      <xsd:attribute name="Revision" type="dt_EDS_Revision" use="required"/>
      <xsd:attribute name="HomeURL" type="dt_EDS_URL" use="optional"/>
      <xsd:attribute name="SpecificationConformance" type="dt_EDS_Char_Array" use="required"/>
   </xsd:attributeGroup>
</xsd:schema>
```

C.2.1.3.3 CIP_Device_Profile.xsd

NOTE This XML schema includes the files “MasterTemplateTypes.xsd” (see C.2.1.3.1) and “CIPDataTypes.xsd” (see C.2.1.3.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <!-- Target namespaces are not specified in this master template -->
   <xsd:redefine schemaLocation="MasterTemplateTypes.xsd">
      <xsd:complexType name="ISO15745Reference_DataType">
         <xsd:complexContent>
            <xsd:restriction base="ISO15745Reference_DataType">
               <xsd:sequence>
                  <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
                  <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
                  <xsd:element name="ProfileTechnology" type="xsd:string" fixed="CIP"/>
               </xsd:sequence>
            </xsd:restriction>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:redefine>
   <xsd:include schemaLocation="CIPDataTypes.xsd"/>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
```

```
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="DeviceIdentity"/>
            <xsd:element ref="DeviceManager" minOccurs="0"/>
            <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
            <xsd:element ref="ApplicationProcess" minOccurs="0"/>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"/>
         </xsd:sequence>
         <xsd:attributeGroup ref="ag_FileDescription"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="DeviceIdentity">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="DeviceIdentity_InstanceAttributes">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="VendCode">
                        <xsd:complexType>
                           <xsd:simpleContent>
                              <xsd:extension base="dt_UINT">
                                 <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                              </xsd:extension>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="VendName">
                        <xsd:complexType>
                           <xsd:simpleContent>
                              <xsd:extension base="dt_EDS_Char_Array"/>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="SpecificationConformance" type="dt_EDS_Char_Array"
minOccurs="0"/>
                     <xsd:element name="ProdType">
                        <xsd:complexType>
                           <xsd:simpleContent>
                              <xsd:extension base="dt_UINT">
                                 <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                              </xsd:extension>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="ProdTypeStr">
                        <xsd:complexType>
                           <xsd:simpleContent>
```

```
                              <xsd:extension base="dt_EDS_Char_Array"/>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="ProdCode">
                        <xsd:complexType>
                           <xsd:simpleContent>
                              <xsd:extension base="dt_UINT">
                                 <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                              </xsd:extension>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="ProdRevision">
                        <xsd:complexType>
                           <xsd:sequence>
                              <xsd:element name="MajRev" type="dt_USINT"/>
                              <xsd:element name="MinRev" type="dt_USINT"/>
                           </xsd:sequence>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Status" minOccurs="0">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="SerialNumber" minOccurs="0">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="ProdName">
                        <xsd:complexType>
                           <xsd:simpleContent>
                              <xsd:extension base="xsd:string">
                                 <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                              </xsd:extension>
                           </xsd:simpleContent>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="State" minOccurs="0">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="ConfigurationConsistencyValue" minOccurs="0">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="HeartbeatInterval" minOccurs="0">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Catalog" type="xsd:string" minOccurs="0"/>
                     <xsd:element name="Icon" type="xsd:string" minOccurs="0"/>
                     <xsd:element name="ExcludeFromAdapterRackConnection" type="xsd:string"
minOccurs="0"/>
                     <xsd:element name="DeviceClassification" minOccurs="0">
                        <xsd:complexType>
                           <xsd:sequence>
                              <xsd:element name="Class" maxOccurs="unbounded">
                                 <xsd:complexType>
                                    <xsd:sequence>
                                       <xsd:element name="MainClass">
                                          <xsd:simpleType>
```

```
                                        <xsd:union>
                                           <xsd:simpleType>
                                              <xsd:restriction base="xsd:NMTOKEN">
                                                 <xsd:enumeration value="ControlNet"/>
                                                 <xsd:enumeration value="DeviceNet"/>
                                                 <xsd:enumeration value="EtherNetIP"/>
                                              </xsd:restriction>
                                           </xsd:simpleType>
                                           <xsd:simpleType>
                                              <xsd:restriction base="et_VendorSpecificKeyword"/>
                                           </xsd:simpleType>
                                        </xsd:union>
                                     </xsd:simpleType>
                                  </xsd:element>
                                  <xsd:element name="SubClass" type="xsd:NMTOKEN" minOccurs="0"
maxOccurs="unbounded"/>
                               </xsd:sequence>
                               <xsd:attribute name="id" use="required">
                                  <xsd:simpleType>
                                     <xsd:restriction base="xsd:ID">
                                        <xsd:pattern value="Class[1-9][0-9]{0,4}"/>
                                     </xsd:restriction>
                                  </xsd:simpleType>
                               </xsd:attribute>
                            </xsd:complexType>
                         </xsd:element>
                      </xsd:sequence>
                   </xsd:complexType>
                </xsd:element>
                <xsd:any namespace="##any"/>
             </xsd:sequence>
          </xsd:complexType>
       </xsd:element>
       <xsd:element name="DeviceIdentity_InstanceOperations" minOccurs="0">
          <xsd:complexType>
             <xsd:sequence>
                <xsd:element name="Get_Attribute_All">
                   <xsd:complexType>
                      <xsd:attribute ref="SupportedService" fixed="true"/>
                   </xsd:complexType>
                </xsd:element>
                <xsd:element name="Reset">
                   <xsd:complexType>
                      <xsd:attribute ref="SupportedService" fixed="true"/>
                   </xsd:complexType>
                </xsd:element>
                <xsd:element name="Get_Attribute_Single">
                   <xsd:complexType>
                      <xsd:attribute ref="SupportedService"/>
                   </xsd:complexType>
                </xsd:element>
                <xsd:any namespace="##any"/>
             </xsd:sequence>
          </xsd:complexType>
       </xsd:element>
       <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
 </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceManager">
 <xsd:complexType>
    <xsd:sequence>
       <xsd:element name="DeviceIdentity_ClassAttributes" minOccurs="0">
          <xsd:complexType>
             <xsd:sequence>
                <xsd:element name="ObjectRevision">
                   <xsd:complexType>
                      <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                   </xsd:complexType>
                </xsd:element>
                <xsd:element name="MaxInstance">
                   <xsd:complexType>
                      <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                   </xsd:complexType>
```

```
                        </xsd:element>
                        <xsd:element name="MaxIDClassAttributes">
                           <xsd:complexType>
                              <xsd:simpleContent>
                                 <xsd:extension base="xsd:string">
                                    <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                                 </xsd:extension>
                              </xsd:simpleContent>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="MaxIDInstanceAttributes">
                           <xsd:complexType>
                              <xsd:simpleContent>
                                 <xsd:extension base="xsd:string">
                                    <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                                 </xsd:extension>
                              </xsd:simpleContent>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="DeviceIdentity_ClassOperations" minOccurs="0">
                  <xsd:complexType>
                     <xsd:sequence>
                        <xsd:element name="Get_Attribute_All">
                           <xsd:complexType>
                              <xsd:attribute ref="SupportedService"/>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="Reset">
                           <xsd:complexType>
                              <xsd:attribute ref="SupportedService"/>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="Get_Attribute_Single">
                           <xsd:complexType>
                              <xsd:attribute ref="SupportedService"/>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="Find_Next_Object_Instance">
                           <xsd:complexType>
                              <xsd:attribute ref="SupportedService"/>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="Modular" minOccurs="0">
                  <xsd:complexType>
                     <xsd:choice>
                        <xsd:element name="Chassis">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element name="DefineSlotsInRack" type="dt_UINT"/>
                                 <xsd:element name="SlotDisplayRule" type="et_ParamReference"
minOccurs="0"/>
                                 <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="Module">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element name="Width" type="dt_UINT"/>
                                 <xsd:element name="Rack" maxOccurs="unbounded">
                                    <xsd:complexType>
                                       <xsd:sequence>
                                          <xsd:element name="VendCode" type="dt_UINT"/>
                                          <xsd:element name="ProdType" type="dt_UINT"/>
                                          <xsd:element name="ProdCode" type="dt_UINT"/>
                                          <xsd:element name="MajRev" type="dt_USINT"/>
```

```
                                    <xsd:element name="MinRev" type="dt_USINT"/>
                                    <xsd:element name="LegalSlot" type="dt_UINT"
maxOccurs="unbounded"/>
                                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                 </xsd:sequence>
                                 <xsd:attribute name="id" use="required">
                                    <xsd:simpleType>
                                       <xsd:restriction base="xsd:ID">
                                          <xsd:pattern value="Rack[1-9][0-9]{0,4}"/>
                                       </xsd:restriction>
                                    </xsd:simpleType>
                                 </xsd:attribute>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:element name="ExternalID" type="dt_EPATH" minOccurs="0"/>
                           <xsd:element name="GenericID" type="dt_EPATH" minOccurs="0"/>
                           <xsd:element name="ExternIDExactMatch" minOccurs="0">
                              <xsd:simpleType>
                                 <xsd:restriction base="xsd:NMTOKEN">
                                    <xsd:enumeration value="Yes"/>
                                    <xsd:enumeration value="No"/>
                                 </xsd:restriction>
                              </xsd:simpleType>
                           </xsd:element>
                           <xsd:element name="Query" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="Path" type="dt_EPATH"/>
                                    <xsd:element name="Service" type="dt_USINT"/>
                                    <xsd:element name="Size">
                                       <xsd:simpleType>
                                          <xsd:restriction base="dt_USINT">
                                             <xsd:minInclusive value="1"/>
                                             <xsd:maxInclusive value="16"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:element>
                                    <xsd:element name="ExternalID" type="dt_EPATH"/>
                                 </xsd:sequence>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                        </xsd:sequence>
                     </xsd:complexType>
                  </xsd:element>
               </xsd:choice>
            </xsd:complexType>
         </xsd:element>
         <xsd:any namespace="##any"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="DeviceFunction">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="ApplicationProcess">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Parameter" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Parameter_Class" minOccurs="0">
                     <xsd:complexType>
                        <xsd:sequence>
                           <xsd:element name="Parameter_ClassAttributes">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="ObjectRevision" minOccurs="0">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
```

```
                                    </xsd:element>
                                    <xsd:element name="MaxInstance">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ParameterClassDescriptor">
                                       <xsd:complexType>
                                          <xsd:simpleContent>
                                             <xsd:extension base="xsd:string">
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                             </xsd:extension>
                                          </xsd:simpleContent>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ConfigurationAssemblyInstance">
                                       <xsd:complexType>
                                          <xsd:simpleContent>
                                             <xsd:extension base="xsd:string">
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                             </xsd:extension>
                                          </xsd:simpleContent>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="NativeLanguage" minOccurs="0">
                                       <xsd:complexType>
                                          <xsd:simpleContent>
                                             <xsd:extension base="xsd:string">
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalSet" use="required"/>
                                             </xsd:extension>
                                          </xsd:simpleContent>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                 </xsd:sequence>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:element name="Parameter_ClassOperations" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="Get_Attribute_All">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Reset">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Get_Attribute_Single">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService" fixed="true"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Set_Attribute_Single">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Restore">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Save">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                 </xsd:sequence>
```

```
                                    </xsd:complexType>
                                 </xsd:element>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="Parameter_Instance" minOccurs="0">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element name="Parameter_InstanceAttributes" minOccurs="0">
                                    <xsd:complexType>
                                       <xsd:sequence>
                                          <xsd:element name="ParameterValue">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Set"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="LinkPathSize">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Set"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="LinkPath">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Set"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="ParamDescriptor">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="DataType">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="DataSize">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="ParameterName">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="UnitsString">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="HelpString">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="MinimumValue">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                          <xsd:element name="MaximumValue">
                                             <xsd:complexType>
                                                <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                             </xsd:complexType>
```

```
                                    </xsd:element>
                                    <xsd:element name="DefaultValue">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ScalingMultiplier">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ScalingDivider">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ScalingBase">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="ScalingOffset">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="MultiplierLink">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="DivisorLink">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="BaseLink">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="OffsetLink">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="DecimalPrecision">
                                       <xsd:complexType>
                                          <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                 </xsd:sequence>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:element name="Parameter_InstanceOperations" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="Get_Attribute_All">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService"/>
                                       </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element name="Get_Attribute_Single">
                                       <xsd:complexType>
                                          <xsd:attribute ref="SupportedService" fixed="true"/>
```

```
                                    </xsd:complexType>
                                </xsd:element>
                                <xsd:element name="Set_Attribute_Single">
                                    <xsd:complexType>
                                        <xsd:attribute ref="SupportedService" fixed="true"/>
                                    </xsd:complexType>
                                </xsd:element>
                                <xsd:element name="Get_Enum_String">
                                    <xsd:complexType>
                                        <xsd:attribute ref="SupportedService"/>
                                    </xsd:complexType>
                                </xsd:element>
                                <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="Param" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base="et_ParamType">
                        <xsd:attribute name="id" use="required">
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:ID">
                                    <xsd:pattern value="Param[1-9][0-9]{0,4}"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:attribute>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="ProxyParam" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base="et_ProxyParamType">
                        <xsd:attribute name="id" use="required">
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:ID">
                                    <xsd:pattern value="ProxyParam[1-9][0-9]{0,4}"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:attribute>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="ProxiedParam" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base="et_ParamType">
                        <xsd:attribute name="id" use="required">
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:ID">
                                    <xsd:pattern value="ProxiedParam[1-9][0-9]{0,4}"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:attribute>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Assembly" minOccurs="0">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Assembly_Class" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="Assembly_ClassAttributes" minOccurs="0">
                            <xsd:complexType>
```

```
                                    <xsd:sequence>
                                       <xsd:element name="ObjectRevision">
                                          <xsd:complexType>
                                             <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Get"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:element name="MaxInstance">
                                          <xsd:complexType>
                                             <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                 </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="Assembly_ClassOperations" minOccurs="0">
                                 <xsd:complexType>
                                    <xsd:sequence>
                                       <xsd:element name="Create">
                                          <xsd:complexType>
                                             <xsd:attribute ref="SupportedService"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:element name="Delete">
                                          <xsd:complexType>
                                             <xsd:attribute ref="SupportedService"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:element name="Get_Attribute_Single">
                                          <xsd:complexType>
                                             <xsd:attribute ref="SupportedService" fixed="true"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                 </xsd:complexType>
                              </xsd:element>
                           </xsd:sequence>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Assembly_Instance" minOccurs="0">
                        <xsd:complexType>
                           <xsd:sequence>
                              <xsd:element name="Assembly_InstanceAttributes" minOccurs="0">
                                 <xsd:complexType>
                                    <xsd:sequence>
                                       <xsd:element name="NumberOfMembers">
                                          <xsd:complexType>
                                             <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalGet" use="required"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:element name="AssemblyMemberList">
                                          <xsd:complexType>
                                             <xsd:attribute name="Access_Rule"
type="at_AccessType_OptionalSet" use="required"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:element name="AssemblyData">
                                          <xsd:complexType>
                                             <xsd:attribute name="Access_Rule"
type="at_AccessType_Mandatory" use="required" fixed="Set"/>
                                          </xsd:complexType>
                                       </xsd:element>
                                       <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                 </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="Assembly_InstanceOperations" minOccurs="0">
                                 <xsd:complexType>
                                    <xsd:sequence>
                                       <xsd:element name="Delete">
                                          <xsd:complexType>
                                             <xsd:attribute ref="SupportedService"/>
```

```
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService" fixed="true"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Set_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Member">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Set_Member">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Insert_Member">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Remove_Member">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="Assem" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
         <xsd:complexContent>
            <xsd:extension base="et_AssemType">
               <xsd:attribute name="id" use="required">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:ID">
                        <xsd:pattern value="Assem[1-9][0-9]{0,4}"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:attribute>
            </xsd:extension>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ProxyAssem" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
         <xsd:complexContent>
            <xsd:extension base="et_AssemType">
               <xsd:attribute name="id" use="required">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:ID">
                        <xsd:pattern value="ProxyAssem[1-9][0-9]{0,4}"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:attribute>
            </xsd:extension>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ProxiedAssem" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
         <xsd:complexContent>
            <xsd:extension base="et_AssemType">
               <xsd:attribute name="id" use="required">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:ID">
```

```
                                    <xsd:pattern value="ProxiedAssem[1-9][0-9]{0,4}"/>
                                 </xsd:restriction>
                              </xsd:simpleType>
                           </xsd:attribute>
                        </xsd:extension>
                     </xsd:complexContent>
                  </xsd:complexType>
               </xsd:element>
               <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="ParameterGroup" minOccurs="0">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="Group" minOccurs="0" maxOccurs="unbounded">
                  <xsd:complexType>
                     <xsd:sequence>
                        <xsd:element name="NameString" type="dt_EDS_Char_Array"/>
                        <xsd:element name="NumberOfMembers" type="dt_UINT"/>
                        <xsd:choice maxOccurs="unbounded">
                           <xsd:element name="ParameterRef" type="dt_UINT"/>
                           <xsd:element name="VariantRef" type="xsd:NMTOKEN"/>
                           <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                        </xsd:choice>
                     </xsd:sequence>
                     <xsd:attribute name="id" use="required">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:ID">
                              <xsd:pattern value="Group[1-9][0-9]{0,4}"/>
                           </xsd:restriction>
                        </xsd:simpleType>
                     </xsd:attribute>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:attribute name="SupportedService" use="required">
   <xsd:simpleType>
      <xsd:restriction base="xsd:boolean">
         <xsd:pattern value="true|false"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:attribute>
<xsd:complexType name="et_ParamType">
   <xsd:sequence>
      <xsd:element name="LinkPathSize" type="dt_USINT" minOccurs="0"/>
      <xsd:element name="LinkPath" type="dt_EPATH" minOccurs="0"/>
      <xsd:element name="ParamDescriptor" type="dt_WORD"/>
      <xsd:element name="DataType">
         <xsd:simpleType>
            <xsd:union memberTypes="dt_USINT dt_EPATH"/>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element name="DataSize" type="dt_USINT"/>
      <xsd:element name="ParameterName" type="dt_EDS_Char_Array"/>
      <xsd:element name="UnitsString" type="dt_EDS_Char_Array"/>
      <xsd:element name="HelpString" type="dt_EDS_Char_Array"/>
      <xsd:element name="MinimumValue" minOccurs="0"/>
      <xsd:element name="MaximumValue" minOccurs="0"/>
      <xsd:element name="DefaultValue" minOccurs="0"/>
      <xsd:element name="ScalingMultiplier" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="ScalingDivider" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="ScalingBase" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="ScalingOffset" type="dt_INT" minOccurs="0"/>
      <xsd:element name="MultiplierLink" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="DivisorLink" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="BaseLink" type="dt_UINT" minOccurs="0"/>
      <xsd:element name="OffsetLink" type="dt_INT" minOccurs="0"/>
      <xsd:element name="DecimalPrecision" type="dt_USINT" minOccurs="0"/>
      <xsd:element name="InternationalParameterName" type="dt_EDS_Char_Array" minOccurs="0"/>
```

```
         <xsd:element name="InternationalEngineeringUnits" type="dt_EDS_Char_Array" minOccurs="0"/>
         <xsd:element name="InternationalHelpString" type="dt_EDS_Char_Array" minOccurs="0"/>
         <xsd:element name="Enum" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence maxOccurs="unbounded">
                  <xsd:element name="EnumValue" type="dt_LINT"/>
                  <xsd:element name="EnumName" type="dt_EDS_Char_Array"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="et_ProxyParamType">
      <xsd:complexContent>
         <xsd:extension base="et_ParamType">
            <xsd:sequence>
               <xsd:element name="ProxyParamSizeAdder" minOccurs="0">
                  <xsd:complexType>
                     <xsd:sequence>
                        <xsd:element name="MinimumValue"/>
                        <xsd:element name="MaximumValue"/>
                        <xsd:element name="DefaultValue"/>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType name="et_AssemType">
      <xsd:sequence>
         <xsd:element name="AssemblyName" type="dt_EDS_Char_Array" minOccurs="0"/>
         <xsd:element name="AssemblyPath" type="dt_EPATH" minOccurs="0"/>
         <xsd:element name="AssemblyDataSize" type="dt_UINT" minOccurs="0"/>
         <xsd:element name="AssemblyDescriptor" type="dt_WORD" minOccurs="0"/>
         <xsd:element name="AssemblyMember" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:choice>
                  <xsd:element name="MemberSize" type="dt_UINT"/>
                  <xsd:element name="MemberReference" type="et_MemberReferenceType"/>
                  <xsd:element name="VariantReference">
                     <xsd:complexType/>
                  </xsd:element>
                  <xsd:choice>
                     <xsd:sequence>
                        <xsd:element name="MemberSize" type="dt_UINT"/>
                        <xsd:element name="MemberReference" type="et_MemberReferenceType"/>
                     </xsd:sequence>
                     <xsd:sequence>
                        <xsd:element name="MemberSize" type="dt_UINT"/>
                        <xsd:element name="VariantReference">
                           <xsd:complexType/>
                        </xsd:element>
                     </xsd:sequence>
                  </xsd:choice>
               </xsd:choice>
            </xsd:complexType>
         </xsd:element>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="et_MemberReferenceType">
      <xsd:union memberTypes="et_AssemReference et_ParamReference dt_UDINT dt_EPATH xsd:NMTOKEN"/>
   </xsd:simpleType>
</xsd:schema>
```

C.2.2 Device profile template description – XML encapsulation of EDS files

C.2.2.1 General

The device profile XML files used to encapsulate EDS files shall comply with the device profile XML schema as specified in C.2.2.2.

The semantics of the sub-elements of the ExternalProfileHandle element, used to reference an existing EDS file, are specified in Table C.2. Depending on the value of the attribute WrapperReference, the EDS file will be referenced using either identification elements from the EDS file itself, or from the product described by this EDS.

NOTE 1 Choice of relevant identification elements will depend upon the expected usage of the wrapper file.

Table C.2 — ExternalProfileHandle elements

XML schema elements	WrapperReference = FILEINFO	WrapperReference =DEVICEINFO
ProfileIdentification	EDS File description text[a]	VendorID, Device Type, Product Code [b]
ProfileRevision	EDS Revision[a]	Product Revision [b]
ProfileLocation	EDS HomeURL[a]	Icon File Name [b]
[a] See C.4.1.4.2 for more details [b] See C.4.1.4.3 for more details		

If present, the DeviceIdentity, DeviceManager, DeviceFunction and ApplicationProcess elements should be compatible with the formats specified in C.2.1.3.3.

NOTE 2 This may be used during a transition phase between the legacy EDS format and the full XML format.

C.2.2.2 XML schema : EDS_Device_Profile_wrapper.xsd

NOTE This XML schema includes the file “MasterTemplateTypes.xsd” (see C.2.1.3.1).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <!-- Target namespaces are not specified in this master template -->
   <xsd:redefine schemaLocation="MasterTemplateTypes.xsd">
      <xsd:complexType name="ISO15745Reference_DataType">
         <xsd:complexContent>
            <xsd:restriction base="ISO15745Reference_DataType">
               <xsd:sequence>
                  <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
                  <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
                  <xsd:element name="ProfileTechnology" type="xsd:string" fixed="EDS"/>
               </xsd:sequence>
            </xsd:restriction>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:redefine>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
```

```
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="DeviceIdentity" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="DeviceManager" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="DeviceFunction" maxOccurs="unbounded">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ApplicationProcess" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:any namespace="##any"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ExternalProfileHandle">
               <xsd:complexType>
                  <xsd:complexContent>
                     <xsd:extension base="ProfileHandle_DataType">
                        <xsd:attribute name="WrapperReference" use="optional" default="FILEINFO">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:NMTOKEN">
                                 <xsd:enumeration value="FILEINFO"/>
                                 <xsd:enumeration value="DEVICEINFO"/>
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                     </xsd:extension>
                  </xsd:complexContent>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

C.3 Communication network profile template description

C.3.1 Communication network profile template description – XML based

C.3.1.1 General

The communication network profile XML files shall comply with the communication network profile XML schema as specified in C.3.1.2.

Contents of this XML schema are derived from the communication network profile class diagrams shown in 6.3.2, and extended with additional elements to allow full description of communication network requirements or capabilities.

C.3.1.2 Semantics of XML schema elements

C.3.1.2.1 ProfileBody

This main element is associated with a set of attributes which provide additional information about the profile file.

The semantics of these attributes is specified in C.4.1.4.2.

C.3.1.2.2 ApplicationLayers

C.3.1.2.2.1 ConnectionManager

This element specifies the supported instance attributes and operations of the Connection Manager Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)), together with a description of the individual connection instances.

The semantics of the Connection, ProxyConnect and ProxiedConnect sub-elements of the ConnectionDescriptions element are specified in C.4.1.4.9 and C.4.1.5.3.3.

C.3.1.2.2.2 MessageRouter

This element specifies the supported instance attributes and operations of the Message Router Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)).

C.3.1.2.3 TransportLayers

C.3.1.2.3.1 ENPhysicalLayer

This element identifies the physical layer. The contents of this element are not detailed in this document.

C.3.1.2.3.2 EtherNetIPLinkObject

This element specifies the supported instance attributes and operations of the EtherNet/IP Link Object (see IEC 61158-4:2003 (Type 2)).

C.3.1.2.3.3 TCPIPInterfaceObject

This element specifies the supported instance attributes and operations of the TCP/IP Interface Object (see IEC 61158-4:2003 (Type 2)).

C.3.1.2.3.4 EncapsulationProtocol

This element specifies the supported attributes and operations associated with encapsulation of application messages into TCP/IP (see IEC 61158-6:2003 (Type 2)).

C.3.1.2.3.5 Ports

This element identifies the device ports which are able to route messages from one link to another link.

The semantics of the Port sub-element of the Ports element are specified in C.4.1.4.10 and C.4.2.2.2.

C.3.1.2.4 NetworkManagement

C.3.1.2.4.1 NM-EtherNetIPLinkObject

This element specifies the supported class attributes and operations of the EtherNet/IP Link Object (see IEC 61158-4:2003 (Type 2)).

C.3.1.2.4.2 NM-TCPIPInterfaceObject

This element specifies the supported class attributes and operations of the TCP/IP Interface Object (see IEC 61158-4:2003 (Type 2)).

C.3.1.2.4.3 NM-ConnectionManager

This element specifies the supported class attributes and operations of the Connection Manager Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)).

C.3.1.2.4.4 NM-MessageRouter

This element specifies the supported class attributes and operations of the Message Router Object (see IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2)).

C.3.1.3 XML schema : ENet_CommNet_Profile.xsd

NOTE This XML schema includes the files “MasterTemplateTypes.xsd” (see C.2.1.3.1) and “CIPDataTypes.xsd” (see C.2.1.3.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <!-- Target namespaces are not specified in this master template -->
   <xsd:redefine schemaLocation="MasterTemplateTypes.xsd">
      <xsd:complexType name="ISO15745Reference_DataType">
         <xsd:complexContent>
            <xsd:restriction base="ISO15745Reference_DataType">
               <xsd:sequence>
                  <xsd:element name="ISO15745Part" type="xsd:positiveInteger" fixed="4"/>
                  <xsd:element name="ISO15745Edition" type="xsd:positiveInteger" fixed="1"/>
                  <xsd:element name="ProfileTechnology" type="xsd:string" fixed="EtherNet/IP"/>
               </xsd:sequence>
            </xsd:restriction>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:redefine>
   <xsd:include schemaLocation="CIPDataTypes.xsd"/>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
```

```
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"
fixed="CommunicationNetwork"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" fixed="CSI"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ApplicationLayers"/>
            <xsd:element ref="TransportLayers"/>
            <xsd:element ref="NetworkManagement" minOccurs="0"/>
         </xsd:sequence>
         <xsd:attributeGroup ref="ag_FileDescription"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ApplicationLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ConnectionManager"/>
            <xsd:element ref="MessageRouter"/>
            <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="TransportLayers">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ENPhysicalLayer" minOccurs="0"/>
            <xsd:element ref="EtherNetIPLinkObject" minOccurs="0"/>
            <xsd:element ref="TCPIPInterfaceObject" minOccurs="0"/>
            <xsd:element ref="EncapsulationProtocol" minOccurs="0"/>
            <xsd:element ref="Ports" minOccurs="0"/>
            <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="NetworkManagement">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="NM-EtherNetIPLinkObject" minOccurs="0"/>
            <xsd:element ref="NM-TCPIPInterfaceObject" minOccurs="0"/>
            <xsd:element ref="NM-ConnectionManager" minOccurs="0"/>
            <xsd:element ref="NM-MessageRouter" minOccurs="0"/>
            <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ConnectionManager">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ConnectionManager_InstanceAttributes" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="OpenReqs">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                        </xsd:complexType>
                     </xsd:element>
```

```
                    <xsd:element name="OpenFormatRejects">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="OpenResourceRejects">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="OpenOtherRejects">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="CloseReqs">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="CloseFormatRejects">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="CloseOtherRejects">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="ConnTimeouts">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="NumConnEntries">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="ConnOpenBits">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="CpuUtilization">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="MaxBuffSize">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="BufSizeRemaining">
                       <xsd:complexType>
                          <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                 </xsd:sequence>
              </xsd:complexType>
           </xsd:element>
```

```
            <xsd:element name="ConnectionManager_InstanceOperations" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="Get_Attribute_All">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Set_Attribute_All">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_List">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Set_Attribute_List">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Set_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Forward_Close">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService" fixed="true"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Unconnected_Send">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Forward_Open">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService" fixed="true"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Connection_Data">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Search_Connection_Data">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Ex_Forward_Open">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Object_Owner">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element ref="ConnectionDescriptions" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
```

```
<xsd:element name="ConnectionDescriptions">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Connection" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:complexContent>
                  <xsd:extension base="et_ConnectionType">
                     <xsd:attribute name="id" use="required">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:ID">
                              <xsd:pattern value="Connection[1-9][0-9]{0,4}"/>
                           </xsd:restriction>
                        </xsd:simpleType>
                     </xsd:attribute>
                  </xsd:extension>
               </xsd:complexContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ProxyConnect" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:complexContent>
                  <xsd:extension base="et_ConnectionType">
                     <xsd:attribute name="id" use="required">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:ID">
                              <xsd:pattern value="ProxyConnect[1-9][0-9]{0,4}"/>
                           </xsd:restriction>
                        </xsd:simpleType>
                     </xsd:attribute>
                  </xsd:extension>
               </xsd:complexContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="ProxiedConnect" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:complexContent>
                  <xsd:extension base="et_ConnectionType">
                     <xsd:attribute name="id" use="required">
                        <xsd:simpleType>
                           <xsd:restriction base="xsd:ID">
                              <xsd:pattern value="ProxiedConnect[1-9][0-9]{0,4}"/>
                           </xsd:restriction>
                        </xsd:simpleType>
                     </xsd:attribute>
                  </xsd:extension>
               </xsd:complexContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:key name="ConnectionName">
      <xsd:selector xpath="Connection|ProxyConnect|ProxiedConnect"/>
      <xsd:field xpath="Name_String"/>
   </xsd:key>
</xsd:element>
<xsd:element name="MessageRouter">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="MessageRouter_InstanceAttributes" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Object_List">
                     <xsd:complexType>
                        <xsd:simpleContent>
                           <xsd:extension base="xsd:string">
                              <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                           </xsd:extension>
                        </xsd:simpleContent>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="MaximumConnectionSupported">
                     <xsd:complexType>
                        <xsd:simpleContent>
                           <xsd:extension base="dt_UINT">
                              <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
```

```
                        </xsd:extension>
                     </xsd:simpleContent>
                  </xsd:complexType>
               </xsd:element>
               <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="MessageRouter_InstanceOperations" minOccurs="0">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="Get_Attribute_All">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="Get_Attribute_List">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="Get_Attribute_Single">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
   </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ENPhysicalLayer">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="EtherNetIPLinkObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="EtherNetIPLinkObject_InstanceAttributes" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="InterfaceSpeed">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="InterfaceFlags">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="PhysicalAddress">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="InterfaceCounters">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="MediaCounters">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"
use="required"/>
                     </xsd:complexType>
                  </xsd:element>
```

```
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="EtherNetIPLinkObject_InstanceOperations" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Get_Attribute_All">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="Get_Attribute_Single">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="Get_and_Clear">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="TCPIPInterfaceObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="TCPIPInterfaceObject_InstanceAttributes" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Status">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="ConfigurationCapability">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="ConfigurationControl">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Set"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="PhysicalLinkObject">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Get"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="InterfaceConfiguration">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Set"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="HostName">
                     <xsd:complexType>
                        <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory"
use="required" fixed="Set"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="TCPIPInterfaceObject_InstanceOperations" minOccurs="0">
```

```
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Get_Attribute_All">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="Set_Attribute_All">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="Get_Attribute_Single">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="Set_Attribute_Single">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="EncapsulationProtocol">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="EncapsulationProtocol_Attributes" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="ProtocolVersion" type="dt_USINT"/>
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="EncapsulationProtocol_Operations" minOccurs="0">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="NOP">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="ListServices">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="ListIdentity">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="ListInterfaces">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="RegisterSession">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="UnRegisterSession">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="SendRRData">
                     <xsd:complexType>
                        <xsd:attribute ref="SupportedService" fixed="true"/>
```

```
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="SendUnitData">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService" fixed="true"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="IndicateStatus">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="Cancel">
                  <xsd:complexType>
                     <xsd:attribute ref="SupportedService"/>
                  </xsd:complexType>
               </xsd:element>
               <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
   </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Ports">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Port" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="PortTypeName">
                     <xsd:simpleType>
                        <xsd:union>
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:NMTOKEN">
                                 <xsd:enumeration value="ControlNet"/>
                                 <xsd:enumeration value="ControlNet_Redundant"/>
                                 <xsd:enumeration value="TCP"/>
                                 <xsd:enumeration value="DeviceNet"/>
                              </xsd:restriction>
                           </xsd:simpleType>
                           <xsd:simpleType>
                              <xsd:restriction base="et_VendorSpecificKeyword"/>
                           </xsd:simpleType>
                        </xsd:union>
                     </xsd:simpleType>
                  </xsd:element>
                  <xsd:element name="PortName" type="dt_EDS_Char_Array" minOccurs="0"/>
                  <xsd:element name="PortObject" type="dt_EPATH" minOccurs="0"/>
                  <xsd:element name="PortNumber" type="dt_UINT"/>
                  <xsd:element name="PortSpecific">
                     <xsd:complexType>
                        <xsd:sequence>
                           <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                        </xsd:sequence>
                     </xsd:complexType>
                  </xsd:element>
                  <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
               <xsd:attribute name="id" use="required">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:ID">
                        <xsd:pattern value="Port[1-9][0-9]{0,4}"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:attribute>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="NM-EtherNetIPLinkObject">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="EtherNetIPLinkObject_ClassAttributes" minOccurs="0">
            <xsd:complexType>
```

```
                <xsd:sequence>
                   <xsd:element name="ObjectRevision">
                      <xsd:complexType>
                         <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory" fixed="Get"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:element name="MaxInstance">
                      <xsd:complexType>
                         <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory" fixed="Get"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
             </xsd:complexType>
          </xsd:element>
          <xsd:element name="EtherNetIPLinkObject_ClassOperations" minOccurs="0">
             <xsd:complexType>
                <xsd:sequence>
                   <xsd:element name="Get_Attribute_All">
                      <xsd:complexType>
                         <xsd:attribute ref="SupportedService"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:element name="Get_Attribute_Single">
                      <xsd:complexType>
                         <xsd:attribute ref="SupportedService"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
             </xsd:complexType>
          </xsd:element>
       </xsd:sequence>
    </xsd:complexType>
 </xsd:element>
 <xsd:element name="NM-TCPIPInterfaceObject">
    <xsd:complexType>
       <xsd:sequence>
          <xsd:element name="TCPIPInterfaceObject_ClassAttributes" minOccurs="0">
             <xsd:complexType>
                <xsd:sequence>
                   <xsd:element name="ObjectRevision">
                      <xsd:complexType>
                         <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory" fixed="Get"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:element name="MaxInstance">
                      <xsd:complexType>
                         <xsd:attribute name="Access_Rule" type="at_AccessType_Mandatory" fixed="Get"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
             </xsd:complexType>
          </xsd:element>
          <xsd:element name="TCPIPInterfaceObject_ClassOperations" minOccurs="0">
             <xsd:complexType>
                <xsd:sequence>
                   <xsd:element name="Get_Attribute_All">
                      <xsd:complexType>
                         <xsd:attribute ref="SupportedService"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:element name="Get_Attribute_Single">
                      <xsd:complexType>
                         <xsd:attribute ref="SupportedService"/>
                      </xsd:complexType>
                   </xsd:element>
                   <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
             </xsd:complexType>
          </xsd:element>
       </xsd:sequence>
    </xsd:complexType>
 </xsd:element>
 <xsd:element name="NM-ConnectionManager">
```

```
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ConnectionManager_ClassAttributes" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="ObjectRevision">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="MaxInstance">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalSet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="OptionalAttributeList">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="ConnectionManager_ClassOperations" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="Get_Attribute_All">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_List">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="NM-MessageRouter">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="MessageRouter_ClassAttributes" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="ObjectRevision">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="OptionalAttributeList">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="OptionalServiceList">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="MaxIDClassAttributes">
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="MaxIDInstanceAttributes">
```

```
                        <xsd:complexType>
                           <xsd:attribute name="Access_Rule" type="at_AccessType_OptionalGet"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="MessageRouter_ClassOperations" minOccurs="0">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="Get_Attribute_All">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_List">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:element name="Get_Attribute_Single">
                        <xsd:complexType>
                           <xsd:attribute ref="SupportedService"/>
                        </xsd:complexType>
                     </xsd:element>
                     <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:attribute name="SupportedService" use="required">
      <xsd:simpleType>
         <xsd:restriction base="xsd:boolean">
            <xsd:pattern value="true|false"/>
         </xsd:restriction>
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:complexType name="et_ConnectionType">
      <xsd:sequence>
         <xsd:element name="Trigger_Transport" type="dt_DWORD"/>
         <xsd:element name="Connection_Parameters" type="dt_DWORD"/>
         <xsd:element name="O-T_RPI" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="dt_UDINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="O-T_Size" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="dt_UINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="O-T_Format" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="et_ParamReference et_AssemReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="T-O_RPI" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="dt_UDINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="T-O_Size" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="dt_UINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="T-O_Format" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="et_ParamReference et_AssemReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Config1_Size" minOccurs="0">
            <xsd:simpleType>
```

```
               <xsd:union memberTypes="dt_UINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Config1_Format" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="et_ParamReference et_AssemReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Config2_Size" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="dt_UINT et_ParamReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Config2_Format" minOccurs="0">
            <xsd:simpleType>
               <xsd:union memberTypes="et_ParamReference et_AssemReference"/>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element name="Name_String" type="dt_EDS_Char_Array"/>
         <xsd:element name="Help_String" type="dt_EDS_Char_Array"/>
         <xsd:element name="Path"/>
         <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

C.3.2 Communication network profile template description – XML encapsulation of EDS files

C.3.2.1 General

The communication network profile XML files used to encapsulate EDS files shall comply with the communication network profile XML schema as specified in C.3.2.2.

The semantics of the sub-elements of the ExternalProfileHandle element, used to reference an existing EDS file, are specified in Table C.2. Depending on the value of the attribute WrapperReference, the EDS file will be referenced using either identification elements from the EDS file itself, or from the product described by this EDS.

NOTE Choice of relevant identification elements will depend upon the expected usage of the wrapper file.

C.3.2.2 XML schema : EDS_CommNet_Profile_wrapper.xsd

NOTE This XML schema includes the file “MasterTemplateTypes.xsd” (see C.2.1.3.1).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <!-- Target namespaces are not specified in this master template -->
   <xsd:redefine schemaLocation="MasterTemplateTypes.xsd">
      <xsd:complexType name="ISO15745Reference_DataType">
         <xsd:complexContent>
            <xsd:restriction base="ISO15745Reference_DataType">
               <xsd:sequence>
                  <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
                  <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
                  <xsd:element name="ProfileTechnology" type="xsd:string" fixed="EDS"/>
               </xsd:sequence>
            </xsd:restriction>
         </xsd:complexContent>
      </xsd:complexType>
   </xsd:redefine>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ProfileHeader"/>
            <xsd:element ref="ProfileBody"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
```

```
   </xsd:annotation>
   <xsd:element name="ProfileHeader">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileName" type="xsd:string"/>
            <xsd:element name="ProfileSource" type="xsd:string"/>
            <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"
fixed="CommunicationNetwork"/>
            <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
            <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
            <xsd:element name="IASInterfaceType" type="IASInterface_DataType" fixed="CSI"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:element name="ProfileBody">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ExternalProfileHandle">
               <xsd:complexType>
                  <xsd:complexContent>
                     <xsd:extension base="ProfileHandle_DataType">
                        <xsd:attribute name="WrapperReference" use="optional" default="FILEINFO">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:NMTOKEN">
                                 <xsd:enumeration value="FILEINFO"/>
                                 <xsd:enumeration value="DEVICEINFO"/>
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                     </xsd:extension>
                  </xsd:complexContent>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

C.4 Electronic Data Sheet (EDS)

C.4.1 Common CIP EDS requirements

C.4.1.1 General

This subclause specifies the file encoding requirements of the Electronic Data Sheet (EDS) which are common to all CIP-based networks. The EDS encoding requirements define the standard file encoding format to use for CIP products without regard to the configuration tool host platform or file system.

The term “file” as used in this chapter refers to any recognized file format associated with a configuration tool’s file system without regard to the file storage media.

An EDS file is defined as an ASCII file, which includes an ASCII representation of objects in the device that can be accessed from the network (e.g. Parameter and Assembly), and some additional information required to support object addressing.

C.4.1.2 EDS content

C.4.1.2.1 EDS structure

A single file shall contain the entire EDS. An EDS shall consist of sections. Table C.3 summarizes the structure of the sections which are common to several CIP-based networks, the corresponding legal section delimiters, and the order of these sections in an EDS.

Table C.3 — CIP EDS file structure

Section name	Legal delimiter	Placement	Required/Optional
File Description	[File]	1	Required
Device Description	[Device]	2	Required
Device Classification	[Device Classification]	[a]	Optional
Parameter Class	[ParamClass]	[a]	Optional
Parameters	[Params]	[a]	Optional
Parameter Groups	[Groups]	[a]	Optional
Assembly	[Assembly]	[a]	Optional
Connection Characteristics	[Connection Manager]	[a]	Optional
Port	[Port]	[a]	Optional
Modular	[Modular]	[a]	Optional
Vendor Specific	[VendorID_vendorspecifickeyword]	Last	Optional
[a] Placement of these optional groups only needs to follow the required groups			

The Electronic Data Sheet (EDS) contents shall be organised as follows:

— all EDS files shall contain the File Description section, which shall be the first section in the EDS file and shall use the legal delimiter [File];

— all EDS files shall contain the Device Description section, which shall immediately follow the File Description section and shall use the legal delimiter [Device];

— the optional sections described in this specification may be present in any order provided that no forward references exist within the EDS file;

— the optional Vendor Specific section(s) shall uses the legal delimiter(s) [VendorID_vendorspecifickeyword] as specified in C.4.1.2.2.11 and shall be placed after all the sections defined in this specification.

C.4.1.2.2 EDS formatting rules

C.4.1.2.2.1 General

An EDS file shall consist of sections, entries, fields, comments and white space. This subclause defines the rules that shall be observed when defining an EDS.

C.4.1.2.2.2 EDS White Space

White space may be used in the EDS file, but shall be ignored by all EDS interpreters when it appears outside of fields and double quoted character arrays.

The EDS interpreter shall treat the following characters as white space characters. These characters, read by the interpreter but not encoded as human-readable characters, designate the presence of blank space in a file.

— Space character

— New line

— Carriage Return

— Linefeed

— Tabs, vertical and horizontal

— Form Feed

— End of File marker

— Comments

C.4.1.2.2.3 Keyword characters

All keywords within an EDS file shall be composed of ASCII characters from the following list:

— upper case letters A through Z;

— lower case letters a through z;

— numerals 0 through 9;

— the special character underscore "_";

— the space character.

The space shall only be used in a section keyword. The space shall only appear internal to a section name, and multiple sequential spaces are invalid.

C.4.1.2.2.4 Sections

The EDS file shall be partitioned into required and optional sections.

C.4.1.2.2.5 Section delimiters

Each section in the EDS shall be properly delimited by a section keyword in square brackets (the Legal Delimiter). The valid section legal delimiters shall be those specified in Table C.3.

C.4.1.2.2.6 Section keywords

Each section keyword is defined to be the text between the beginning of section keyword delimiter "[" and the terminating delimiter "]". The characters valid for use in section keywords are defined in C.4.1.2.2.3. There are two types of section keywords, public and vendor specific.

C.4.1.2.2.7 Section order

Each required section shall be placed in the required order as specified in C.4.1.2. Optional sections may be omitted entirely or included with empty data place holders. Except for the Vendor Specific section(s), optional sections may be placed in any order. The Vendor Specific section(s) shall be placed last in the EDS file.

C.4.1.2.2.8 Entry

Each section in the EDS shall contain one or more entries beginning with an entry keyword followed by an equal sign. The entry keyword meaning shall be global in scope, allowing keywords defined in one section to be used in another. Each entry shall terminate with a semicolon. An entry may extend over multiple lines, as long as commas properly delimit the fields.

C.4.1.2.2.9 Entry keywords

An entry keyword shall consist of a unique sequence of keyword characters, as defined in C.4.1.2.2.3. There are two types of entry keywords, public and vendor specific.

C.4.1.2.2.10 Public keyword

A Public Keyword shall always be defined within the CIP specification by the responsible vendors associations. A public keyword shall never begin with any numeric digit.

C.4.1.2.2.11 Vendor-specific keywords

Keywords may be vendor-specific. These keywords shall begin with the Vendor ID of the company making the addition followed by an underscore (VendorID_VendorSpecificKeyword). The VendorID shall be displayed in decimal and shall not contain leading zeroes. Each vendor is responsible for maintaining and documenting their vendor-specific keywords.

C.4.1.2.2.12 Entry fields

Each entry shall contain one or more fields. Comma delimiters shall separate all fields. The meaning of the field(s) shall depend on the context of the section. Entry fields are either required or optional, as defined by this specification. A white space or nothing between commas shall be used for optional fields not provided. A semicolon may be used to designate the absence of trailing optional fields. The term “Field Number” shall indicate field position within the entry. Fields shall be numbered from left-to-right (or up-to-down) starting with the number 1.

C.4.1.2.2.13 Field keywords

A field keyword shall consist of a unique sequence of keyword characters, as defined in C.4.1.2.2.3. There are two types of field keywords, public and vendor specific.

C.4.1.2.2.14 Complex data fields

Certain entry fields shall be specified with data that cannot be specified by a single value between the comma delimiters. The ability to further delimit an entry field is defined via the use of one or more sets of matching brace characters "{" and "}". The content between brace characters shall be considered a single item or entry. Content may be grouped in multiple braces.

C.4.1.2.2.15 Comments

Comments shall be delimited with the dollar sign character ($) and the new line character. The EDS interpreter shall treat all characters between the comment delimiters as white space. The $ comment delimiter appearing inside of a field or double quoted character array shall not be treated as a comment delimiter.

EXAMPLE

Some example comments are:

```
$ This is a valid comment line    <NL>
1, 2, 3;                          $ This is a valid comment <NL>
$ Comments cannot span <NL>
more than one line <NL>           <= This is an error - no $
```

C.4.1.2.2.16 EDS formatting structure example

Figure C.1 examples highlight the structure of the Electronic Data Sheet.

```
[section name]
$ Comment - extends to end of line
Entry1=Field1, Field2, Field3;          $ Entire entry on one line
Entry2=Field1, Field2, Field3, Field4;  $ Entire entry on one line

Entry3=                                $ Multiple line entry
       Field1,                         $ Field1
       Field2,                         $ Field2
       Field3;                         $ Field3

Entry4=                                $ Combination
       Field1, Field2,                 $ Fields 1 and 2 on one line
       Field3,                         $ Field3
       Field4;                         $ Field4

Entry5=   1,                           $ Field 1 specifies the value 1
       {1,2,3};                        $ Field 2 specifies an array or
                                       $ structure with three values

Entry6=   { 44, {22,33,11} };          $ Entry 6 specifies a single field.
                                       $ The field contains two sets of data.
                                       $ The first set is the single value 44
                                       $ The second set contains three values

65535_Entry=                           $ Vendor Specific entry for
          Field1, Field2;              $ Vendor_ID 65535 with two fields
```

Figure C.1 — EDS formatting structure example (informative)

C.4.1.2.3 File naming requirements

No file naming conventions are specified for disk-based EDS files, except for files in a DOS/Windows environment : these files shall have the suffix ".EDS" appended to the file name.

C.4.1.3 EDS data encoding requirements

C.4.1.3.1 General

This section specifies the data encoding requirements for the EDS file.

The information contained in the EDS file may represent attributes of object instances in the device to be configured. All data in the EDS file shall be ASCII text whereas the object class and instance attributes need not be ASCII (available data types are defined in the CIP specification). Therefore, translation between data contained in an EDS file and the object attributes may be needed: this translation is specified in the following subclauses.

The elementary data types specified in the CIP specification are also used for other elements of the EDS, however, the meaning is transformed as described in the following subclauses (see C.4.1.3.3 to C.4.1.3.10).

Some data types are used solely in EDS files (see C.4.1.3.11 to C.4.1.3.14).

C.4.1.3.2 ASCII character file convention

All data in the EDS shall be encoded using 8-bit ASCII characters, where all references to "ASCII characters" mean an 8-bit ASCII character format (as defined by Tables 1 and 2, Row 00 of ISO/IEC 10646-1:2000). Characters that cannot be displayed on an ANSI terminal shall not be used in identifier names or in data representations. The valid ASCII character values shall include newline, tab, and those from 32 to 126 decimal.

C.4.1.3.3 Character string convention – EDS_Char_Array

C.4.1.3.3.1 General

All string data in the EDS file shall be character strings of fixed length, without null terminators, and shall be enclosed by double quotes (EDS_Char_Array data type).

There are two forms of string data conversions. Characters contained between double quotes shall be converted into 8-bit ASCII characters. Characters contained between double quotes that are preceded by a capital L shall be converted into UNICODE (16-bit) characters.

EXAMPLE 1 "This results in a string composed by 8-bit characters"

EXAMPLE 2 L"A string of UNICODE characters, including the Greek character Pi \u03C0"

NOTE The text \u03C0 specifies a single 16-bit character whose value is 03C0. In the UNICODE character set, this is Table 9, Row 3, Basic Greek – the character for lower case "Pi". Descriptions of character escape sequences is described in C.4.1.3.3.5.

C.4.1.3.3.2 Handling insufficient characters in a string field

An EDS interpreter shall use right-justification of characters in a field and fill any unspecified characters with leading blanks (ASCII 0x20) for the remaining length of the string.

EXAMPLE If a parameter has a maximum string length of 8 and receives the string "123AB", the string is interpreted as "~~~123AB", where the tilde characters (~) represent spaces.

C.4.1.3.3.3 Handling excess characters in a string field

If a given string field contains too many characters, the EDS interpreter shall truncate characters from left to right.

EXAMPLE If a parameter has a maximum string length of 8 and receives the string ”I23ABCDEFG”, the string is truncated and interpreted as ”I23ABCDE”.

C.4.1.3.3.4 String concatenation

Multiple strings with no intervening commas shall be concatenated.

EXAMPLE 1
The line : ”ABC” ”123” ”XYZ”
is interpreted as : ”ABC123XYZ”

The strings may also be on separate lines.

EXAMPLE 2
The following lines :
”ABC” $this is a comment
”123”
”XYZ”
are also interpreted as : ”ABC123XYZ”

For a UNICODE string (long string), only the first double quotes mark shall be preceded by a capital L.

EXAMPLE 3 L”ABC” ”123” ”XYZ” is the same as L”ABC123XYZ”.

C.4.1.3.3.5 String escape sequences

The EDS interpreter shall recognize all escape sequences listed in Table C.4. Interpretation is application specific.

Table C.4 — String escape sequences

Escape sequence	Translation
\\	\
\n	newline
\t	tab
\v	vertical tab
\b	backspace
\r	carriage return
\f	form feed
\a	the BELL character (0x07)
\”	”
\’	’
\xnn	single byte containing the value of “nn” as expressed in hexadecimal
\unnnn	pair of bytes containing the value of “nnnn” as expressed in hexadecimal. This form of string escape is only valid where the resultant string data is 16-bits in length, e.g. the L" form of string specification.

If a sequence not listed above is encountered, the interpreting device shall reject the entire string and indicate an error. EDS files shall only contain escape sequences defined in Table C.4.

C.4.1.3.4 ASCII string convention (STRING, SHORT_STRING, STRING2)

All string data types (STRING, SHORT_STRING, STRING2) used in the object attributes shall be converted into EDS_Char_Array in the EDS file.

C.4.1.3.5 STRINGI

The CIP International String (STRINGI) data type is encoded in an EDS file as a complex data representation. The entire content of a STRINGI entry shall be enclosed in a pair of braces. The number of language members, specified as a USINT, shall be followed by the language members definitions, each being enclosed in a pair of braces, and separated by a comma. Each language member of a STRINGI entry shall be specified as four fields. The first field (the language selection) shall be expressed as exactly a three character fixed length string enclosed in double quotation marks – language code as defined in ISO 639-2/T. The string data type shall be expressed using the data type code as defined in the CIP specification for STRING, STRING2, STRINGN or SHORT_STRING. The character set selection shall be expressed as a UINT – as defined in IANA MIB Printer Codes (RFC 1759). The string content portion of the language member shall be expressed as a string or long string.

EXAMPLE

The following represents a STRINGI entry with three languages:

```
Field1 =    { 3,
              {"eng",0xD0,4,"This is an ASCII English language string"},
              {"spa",0xD5,1000,L"Españoles palabras"},        $ “Spanish words”
                                                             $ using UNICODE
              {"deu",0xD0,4,"Spanische Wörter auf Deutsch"}  $“Spanish words in German”
              };
```

C.4.1.3.6 CIP path (EPATH)

The CIP EPATH data type, used in particular to define CIP path strings, shall be encoded in EDS files using the base format defined in ISO 15745-4 for EDS_Char_Array. In addition, the string contents for a CIP path or other EPATH data shall consist of groups of two adjacent hexadecimal characters separated by spaces. Both upper and lower case may be used.

EXAMPLE 1 “20 04 24 01”

EXAMPLE 2 “20 05 24 02 30 04”

C.4.1.3.7 ASCII unsigned integer convention - (USINT, UINT, UDINT, ULINT)

The unsigned integer data types represent positive integer values. Unsigned integer data shall be entered either in decimal or in hexadecimal notation with no whitespace or commas between characters. If hexadecimal notation is used to represent the unsigned integer characters, the two character sequence 0x, with no white space, shall precede the unsigned integer characters.

The range of legal USINT data is:

Decimal Notation: 0 to 255

Hexadecimal Notation: 0x0 to 0xFF

The range of legal UINT data is:

Decimal Notation: 0 to 65535

Hexadecimal Notation: 0x0 to 0xFFFF

The range of legal UDINT data is:

Decimal Notation: 0 to 4294967295

Hexadecimal Notation: 0x0 to 0xFFFFFFFF

The range of legal ULINT data is:

Decimal Notation: 0 to 18446744073709551615

Hexadecimal Notation: 0x0 to 0xFFFFFFFFFFFFFFFF

Leading zeros shall not be used for the decimal notation, but may be used for the hexadecimal notation. For hexadecimal notation, both upper and lower case may be used, and the total number of characters shall be limited to 10 (0x plus 8 more), or 18 (0x plus 16 more) for the ULINT type.

EXAMPLE The decimal UINT value 254 may be represented as 254 (decimal), or as 0xFE (hexadecimal), or as 0x000000FE (hexadecimal) but 0254 (decimal) and 0x0000000FE (hexadecimal) are illegal.

C.4.1.3.8 ASCII signed integer convention - (SINT, INT, DINT, LINT)

The SINT, INT, DINT and LINT data types represent signed integer data values. Signed integer data shall be entered either in decimal or in hexadecimal notation with no whitespace or commas between characters. If hexadecimal notation is used to represent the signed integer characters, the two character sequence 0x, with no white space, shall precede the integer value characters.

The range of legal SINT data is:

Decimal Notation: -128 to 127

Hexadecimal Notation: 0x80 to 0x7F

The range of legal INT data is:

Decimal Notation: -32768 to 32767

Hexadecimal Notation: 0x8000 to 0x7FFF

The range of legal DINT data is:

Decimal Notation: -2147483648 to 2147483647

Hexadecimal Notation: 0x80000000 to 0x7FFFFFFF

The range of legal LINT data is:

Decimal Notation: -9223372036854775808 to 9223372036854775807

Hexadecimal Notation: 0x8000000000000000 to 0x7FFFFFFFFFFFFFFF

Leading zeros shall not be used for the decimal notation, but may be used for the hexadecimal notation. For hexadecimal notation, both upper and lower case may be used, and the total number of characters shall be limited to 10 (0x plus 8 more), or 18 (0x plus 16 more) for the LINT type.

EXAMPLE The decimal INT value 254 may be represented as 254 (decimal), or as 0xFE (hexadecimal), or as 0x000000FE (hexadecimal) but 0254 (decimal) and 0x0000000FE (hexadecimal) are illegal.

C.4.1.3.9 ASCII word convention - (BYTE, WORD, DWORD, LWORD)

The BYTE, WORD, DWORD and LWORD data types represent bit-addressable values. These values are considered discrete bit position values and are not intended to represent either signed or unsigned integer values. However, these values, for convenience, shall be entered either in decimal, hexadecimal or binary notation with no whitespace or commas between characters. If hexadecimal (respectively binary) notation is used to represent value characters, the two character sequence 0x (respectively 0b), with no white space, shall precede the value characters.

The range of legal BYTE data is:

Decimal Notation: 0 to 255

Hexadecimal Notation: 0x0 to 0xFF

Binary Notation: 0b00000000 to 0b11111111

The range of legal WORD data is:

Decimal Notation: 0 to 65535

Hexadecimal Notation: 0x0 to 0xFFFF

Binary Notation: 0b0000000000000000 to 0b1111111111111111

The range of legal DWORD data is:

Decimal Notation: 0 to 4294967295

Hexadecimal Notation: 0x0 to 0xFFFFFFFF

Binary Notation: 0b00000000000000000000000000000000 to

0b11111111111111111111111111111111

The range of legal LWORD data is:

Decimal Notation: 0 to 18446744073709551615

Hexadecimal Notation: 0x0 to 0xFFFFFFFFFFFFFFFF

Binary Notation: 0b0000000000000000000000000000000-

00000000000000000000000000000000 to

0b1111111111111111111111111111111-

11111111111111111111111111111111

Leading zeros shall not be used for the decimal notation, but may be used for the hexadecimal and binary notations. For hexadecimal notation, both upper and lower case may be used, and the total number of characters shall be limited to 10 (0x plus 8 more), or 18 (0x plus 16 more) for the LWORD type.

EXAMPLE The decimal WORD value 254 may be represented as 254 (decimal), or as 0xFE (hexadecimal), or as 0x000000FE (hexadecimal) but 0254 (decimal) and 0x0000000FE (hexadecimal) are illegal.

C.4.1.3.10 ASCII floating point convention (REAL, LREAL)

The REAL and LREAL data types represent binary floating point values. The internal representation of these data formats are described by the IEEE Standard 754. This standard describes both numeric values and bit sequences which are interpreted as “Not a Number” (NaN) symbolic values and positive and negative infinity. The floating point values may be entered as either integer values, values based upon decimal floating point representation, or values entered in “scientific” notation using a base value and an offset in exponential form. Integer values are the same as those shown for the INT, DINT or LINT data types. These values cannot be used to represent fractional values. Decimal floating point values are those which have both a whole integer and fractional component. The whole value and fractional components are separated by a decimal point “.” or period character. The exponential (scientific) notation form of the value is the same as the fractional value representation with the addition of an exponential component. This exponent is always a signed integer power to ten applied to the base value.

NOTE The maximum precision of a floating point value is determined by the capabilities of the internal binary format, i.e. the number of binary digits available to encode the mantissa. Therefore, using a large number of decimal digits within the decimal notation (or the mantissa part of the scientific notation) of a floating point value is more for presentation convenience than precision. EDS defines arbitrary limits for the number of decimal digits.

The range of legal REAL data (single IEEE, 32 bit format) is based upon the formula:

$$value = (-1)^{s} \bullet (2)^{e-127} \bullet (m)$$

Where:

— "s" is the value of the sign bit;

— "e" is the eight bit exponent. This exponent allows an exponent range between -126 and +127;

— "m" is the normalized 24 bit mantissa (23 internal to the storage plus one hidden bit). This allows a range of mantissa values to range between 0 and 16777215.

The combination of "e" and "m" allows an approximate absolute value range of 0 to $3{,}4028e^{38}$.

EDS uses for REAL data the following floating point values notations:

Integer (Fixed) Notation: -16777215 to 16777215

Decimal (Floating Point) Notation: 0.0 to ±9999999999999999

Where the total number of digits shall not exceed 16, in addition to the decimal point and sign characters. Both the decimal point character and the sign character may be omitted (+ sign is implied if the sign character is omitted).

Scientific Notation: 0.0 to ±nn.nnnnnnnnnE±xxxx

Where the total number of digits in the mantissa shall not exceed 11 (in addition to the decimal point character and sign character), and the number of digits in the exponent shall not exceed 4 (in addition to the "E" character and sign character). The decimal point may be placed anywhere in the mantissa. Both the decimal point character and the sign character may be omitted in the mantissa (+ sign is implied if the sign character is omitted).

The range of legal LREAL data (double IEEE, 64 bit format) is based upon the formula:

$$value = (-1)^{s} \bullet (2)^{e-1023} \bullet (m)$$

Where:

— "s" is the value of the sign bit;

— "e" is the eleven bit exponent. This exponent allows an exponent range between -1022 and +1023;

— "m" is the normalized 53 bit mantissa (52 internal to the storage plus one hidden bit). This allows a range of mantissa values to range between 0 and 9007199254740991.

The combination of "e" and "m" allows an approximate absolute value range of 0 to $1{,}7976e^{308}$.

EDS uses for LREAL data the following floating point values notations:

Integer (Fixed) Notation: -9007199254740991 to 9007199254740991

Decimal (Floating Point) Notation: 0.0 to ±9999999999999999

Where the total number of digits shall not exceed 16, in addition to the decimal point and sign characters. Both the decimal point character and the sign character may be omitted (+ sign is implied if the sign character is omitted).

Scientific Notation: 0.0 to ±nnnn.nnnnnnnnnnnnE±xxxx

Where the total number of digits in the mantissa shall not exceed 16 (in addition to the decimal point character and sign character), and the number of digits in the exponent shall not exceed 4 (in addition to the "E" character and sign character). The decimal point may be placed anywhere in the mantissa. Both the decimal point character and the sign character may be omitted in the mantissa (+ sign is implied if the sign character is omitted).

In addition to the above value entries, the floating point representation allows for two styles of "Not a Number" or NaN symbolic entries, and two forms of infinity. There are two types of NaN; a Signaling NaN and a Quiet NaN. Also, the format allows for the representation of the values positive and negative infinity. For these cases, the following special words are reserved and shall be used to represent the entry of the associated floating point symbol:

— Quiet Not a Number: QUIET-NAN

— Signaling Not a Number: SIGNAL-NAN

— Positive Infinity: INFINITY (or +INFINITY)

— Negative Infinity: -INFINITY

C.4.1.3.11 EDS_Date

The EDS_Date data type shall be of the format mm-dd-yyyy, where mm is the month, dd is the day of the month and yyyy is the year. Valid values for the month, day and year portions of the mm-dd-yyyy shall be:

— mm 01 through 12;

— dd 01 through 31 (depending upon the month and year);

— yyyy 1996 through 9999.

Two character years representations may be used in which case the EDS_Date data type shall be of the format; mm-dd-yy, where mm is the month, dd is the day of the month and yy is the year. In this case, the two digits for the year have an implied leading 19, such that yy=96 shall represent the year 1996. Valid values for the month, day and year portions of the mm-dd-yy parameters shall be:

— mm 01 through 12;

— dd 01 through 31 (depending upon the month and year);

— yy 96 through 99 (a leading 19 is implied).

NOTE Two character year representations are not recommended.

C.4.1.3.12 EDS_Time_Of_Day

The EDS_Time_Of_Day data type shall be of the format hh:mm:ss, where hh is hours, mm is minutes and ss is seconds. Valid values for the hours, minutes and seconds shall be:

— hh 00 through 23.

— mm 00 through 59;

— ss 00 through 59.

C.4.1.3.13 EDS_Revision

The EDS_Revision data type shall be of format Major_Revision.Minor_Revision with valid values of:

— Major_Revision 0 to 9;

— Minor_Revision 0 to 9.

An EDS_Revision of 0.0 shall be invalid.

EXAMPLE An EDS_Revision of 1.4 corresponds to a major revision of 1 and a minor revision of 4.

C.4.1.3.14 EDS_URL Uniform Resource Locator

All references to EDS_URL within the EDS requirements are for the formalized information necessary to locate and access resources via an Internet capable mechanism. An EDS_URL shall be encoded in EDS files using the base format defined in ISO 15745-4 for EDS_Char_Array. In addition, the string contents for an EDS_URL shall follow the format defined by the Internet's Network Working Group RFC 1738 "Uniform Resource Locator (URL)". For specifications made within an EDS file, the EDS_URL shall be limited to any of the forms:

— http;

— ftp;

— file.

C.4.1.4 Basic EDS file requirements

C.4.1.4.1 Overview

This subclause describes the basic sections of an EDS which are common to several CIP-based networks, and specifies the corresponding usage requirements. Table C.5 gives the subclause location of these section definitions.

Table C.5 — Basic sections definition

EDS sections	Defined in
File description section	C.4.1.4.2
Device description section	C.4.1.4.3
Device classification section	C.4.1.4.4
Parameter class section	C.4.1.4.5
Parameters section	C.4.1.4.6
Parameter groups section	C.4.1.4.7
Assembly section	C.4.1.4.8
Connection manager section	C.4.1.4.9
Port section	C.4.1.4.10
Modular section	C.4.1.5.2

C.4.1.4.2 File Description section

The file description section shall contain administrative information about the EDS file. A configuration tool shall read this information, format it, and display it to the user. The user can also access this section with a text file viewer and display the unformatted information. This section shall not require modification unless the user manually modifies the file. The file description section shall contain the entries shown in Table C.6.

Table C.6 — File description format

Entry name	Entry keyword	Field number	Data type	Required/Optional
File Description Text	DescText	1	EDS_Char_Array	Required
File Creation Date	CreateDate	1	EDS_Date	Required
File Creation Time	CreateTime	1	EDS_Time_Of_Day	Required
Last Modification Date	ModDate	1	EDS_Date	Conditional
Last Modification Time	ModTime	1	EDS_Time_Of_Day	Conditional
EDS Revision	Revision	1	EDS_Revision	Required
Home URL	HomeURL	1	EDS_URL	Optional

The entries in the file description section shall provide the information as shown in Table C.7.

Table C.7 — File description entries

Entries	Description
File Description Text	A single line of text displayed by the configuration tool. The EDS developer shall assign a meaningful line of text for this entry. Double quotes shall enclose all character arrays.
File Creation Date	The creation date of the EDS, assigned by the EDS developer. Provided only for convenience, this date can be used to get version information about the file. A configuration tool shall not use this information to perform any type of version control, but it may display the contents.
File Creation Time	The creation time of the EDS, assigned by the EDS developer. Provided only for convenience, this time can be used to get version information about the file. A configuration tool shall not use this information to perform any type of version control, but it may display the contents.
Last Modification Date	The date of the last modification to the EDS. A configuration tool that allows modification of the EDS file shall update this field as needed. Provided only for convenience, the configuration tool shall display the contents of this entry if it exists. If a configuration tool changes the EDS, the configuration tool shall update this field. However, if the EDS is modified manually or with a text editor, this field shall also be updated. This entry is required if either : – the EDS file is modified by a software tool – the Last Modification Time entry is present
Last Modification Time	The time of the last modification to the EDS. A configuration tool that allows modification of the EDS file shall update this entry as needed. Provided for convenience, the configuration tool shall display the contents of this entry if it exists. If a configuration tool changes the EDS, the configuration tool shall update this field. However, if the EDS is modified manually or with a text editor, this field shall also be updated.
EDS Revision	The revision of the EDS. The EDS revision need not have any relationship to the product's revision, it is simply the revision of the EDS file itself.
Home URL	Uniform Resource Locator of the master EDS file, the Icon file and other files related to this EDS. The HomeURL shall specify a complete qualified URL for referencing a master version of the EDS file. In addition, the referenced area (without the file name specification) is used to specify an area where other related file(s) relating to the device described by this EDS are contained.

Figure C.2 is an example that shows a typical [File] section.

```
[File]
   DescText = "Smart Widget EDS File";
```

```
CreateDate = 04-03-94;                  $ created
CreateTime = 17:51:44;
ModDate = 04-06-94;                     $ last changed
ModTime = 22:07:30;
Revision = 2.1;                         $ Revision of EDS
HomeURL = "http://www.odva.org/EDS/example.eds";
```

Figure C.2 — [File] section example (informative)

C.4.1.4.3 Device Description section

The Device Description section shall contain manufacturer's information about the device, including some of the same values as in a device Identity Object. The device description section shall contain the entries specified in Table C.8

Table C.8 — Device description format

Entry name	Entry keyword	Field number	Data type	Required/Optional
Vendor Id[a,b]	VendCode	1	UINT	Required
Vendor Name	VendName	1	EDS_Char_Array	Required
Device Type[a,b]	ProdType	1	UINT	Required
Device Type String	ProdTypeStr	1	EDS_Char_Array	Required
Product Code[a,b]	ProdCode	1	UINT	Required
Major Revision[a,b]	MajRev	1	USINT	Required
Minor Revision[a]	MinRev	1	USINT	Required
Product Name[c]	ProdName	1	EDS_Char_Array	Required
Catalog Number	Catalog	1	EDS_Char_Array	Optional
Exclude from Adapter Rack Connection	ExcludeFromAdapterRackConnection	1	EDS_Char_Array	Optional
Icon File Name	Icon	1	EDS_Char_Array	Optional

[a] This entry represents an attribute of the Identity Object

[b] This entry is used to match an EDS with a specific product/revision

[c] This entry represents an attribute of the Identity Object, although the data type may be slightly different.

The entry name for the device description field describes the unique data entry line number.

A configuration tool shall use the required entries in the device description section to match the EDS to the device being configured. The entries in the device description section shall provide the information as shown in Table C.9.

Table C.9 — Device description entries

Entries	Description
Vendor ID	Numeric vendor identifier as defined by the Identity Object, Attribute 1.
Vendor Name	Textual vendor name. When displayed, truncation may occur to meet the display capabilities.
Device Type	Numeric device identifier as defined by the Identity Object, Attribute 2.
DeviceType String	Textual description of device type exactly as defined in the corresponding CIP device profile. The individual vendors may choose the strings for vendor specific device types.
Product Code	Vendor assigned numeric product code identifier as defined by the Identity Object, Attribute 3. Each product code shall have its own EDS.
Major Revision	Vendor-assigned major revision number as defined by the Identity Object, Attribute 4. The major revision of a product may be typically incremented when there is a change to the form, fit, or function of the device. Changes to major revisions shall be used by a configuration tool to match a device to an EDS.
Minor Revision	Vendor-assigned minor revision number as defined by the Identity Object, Attribute 4. The minor revision number shall be used to identify changes in a product that do not effect user configuration choices (e.g. firmware bug fixes, an additional LED, internal hardware changes). Changes in minor revisions shall not be used by a configuration tool to match a device with an EDS.
Product Name	Textual product name as defined by the Identity Object, Attribute 7. When displayed, truncation may occur to meet the display capabilities.
Catalog Number	Textual catalog or model number. One or more catalog numbers may be associated with a particular product code. NOTE In the case of multiple catalog numbers, it is still useful to provide as much of the catalog number as is practical. For example, 1438-BAC7xx where 'xx' represents variants in the catalog number supported by this product code/EDS.
ExcludeFromAdapterRackConnection	This field is used to describe if a rack based device is required to be excluded from an adapter rack connection. If the field value is the string "Yes" this module shall be excluded from adapter rack connections by resetting the associated slot mask bits (input, output and configuration). If the field value is the string "No" or this optional field is omitted the associated slot mask bits may be set.
Icon File Name	File name of an icon file. Identifies a file that contain a graphical representation of the device. The file shall have the *.ICO MSWindows format, and shall minimally contain a 16x16 icon. The file may also contain 32x32, 48x48, and 64x64 icons. The location of the icon file is the combination of the location specified by the HomeURL keyword (without the HomeURL file name component) and the file name specified by this keyword. This keyword shall only be present when a HomeURL keyword exists.

Figure C.3 is an example that shows a typical Device Section.

```
[Device]
   VendCode = 65535;
   VendName = "Widget-Works, Inc.";
   ProdType = 0;
   ProdTypeStr = "Generic";
   ProdCode = 42;
   MajRev = 1;            $ Device Major Revision
   MinRev = 1;            $ Device Minor Revision
   ProdName = "Smart-Widget";
   Catalog = "1499-DVG";
   Icon = "example.ico";
```

Figure C.3 — [Device] section example (informative)

C.4.1.4.4 Device Classification section

The Device Classification section shall classify the device described by the EDS into one or more categories of devices. The entry keyword for all classifications shall consist of the character array, "Class", combined with a decimal number. The numbers shall start at 1 for the first class, and shall be incremented for each additional class.

The number of fields for each classification entry shall be variable to allow a tree classification structure similar to a file systems directory structure. Sub-classification of the public classifications shall be reserved. Vendor-specific classifications may be sub-classified at the discretion of the vendor. The first field shall represent the highest level in the tree structure and shall be one of the following field keywords:

— ControlNet;

— DeviceNet;

— EtherNetIP;

— a vendor-specific field keyword.

The vendor-specific field keyword shall begin with the Vendor ID of the company making the addition followed by an underscore (VendorID_VendorSpecificField). The VendorID shall be displayed in decimal and shall not contain leading zeroes. Each vendor is responsible for maintaining and documenting their vendor-specific field keyword.

C.4.1.4.5 Parameter Class section

The parameter class section shall identify general attributes of the configuration parameters described by the EDS, which correspond to a subset of the Parameter Object class attributes as defined in the CIP Object Library.

The parameter class section shall contain the entries specified in Table C.10.

Table C.10 — Parameter class format

Entry name	Entry keyword	Field number	Data type	Required/Optional
Max Instances	MaxInst	1	UINT	Required
Parameter Class Descriptor	Descriptor	1	WORD	Required
Configuration Assembly Instance	CfgAssembly	1	UINT	Required

The entries in the parameter class section shall provide the information as shown in Table C.11.

Table C.11 — Parameter class entries

Entries	Description
Max Instances	Identifies the total number of configuration parameters contained in the device associated with the EDS
Parameter Class Descriptor	Contains bit flags that describe the behavior of the device's parameter objects
Configuration Assembly Instance	Specifies the instance number of the Assembly Object that contains the device configuration data.

The Parameter Class Descriptor entry shall contain bits to describe parameter characteristics as defined in Table C.12. Bits not defined in Table C.12 shall not be used and shall be set to zero (0).

Table C.12 — Parameter class descriptor bit values

Bit	Name	Bit value and meaning
0	Supports individual parameter access	0 = NO parameter can be individually accessed. Only the Configuration assembly is used. 1 = Parameters can be individually accessed.
1	Supports full attributes	0 = Only the current value of a parameter is available within the device. 1 = All configuration data for a parameter is available within the device itself.
2	Non-volatile storage save command	0 = Parameters saved automatically. 1 = Parameters not saved automatically. Need to execute non-volatile storage save command when desired parameters to be saved in non-volatile storage.
3	Params are stored in non-volatile storage	0 = Parameters are not stored in non-volatile storage. 1 = All full parameters are stored in non-volatile storage.

Figure C.4 is an example that shows a typical Parameter Class Section.

```
[ParamClass]
   MaxInst = 3;
   Descriptor = 0x0E;
   CfgAssembly = 3;
```

Figure C.4 — [ParamClass] section example (informative)

C.4.1.4.6 Parameters section

The parameter section shall identify the configuration parameters in a device. The entry keyword shall be one of the following character arrays, “Param”, “ProxyParam”, “ProxiedParam”, combined with a parameter instance number (decimal) for the device, e.g. “Param1”. The actual parameter object instance may, but need not be, implemented in the device. Conversely, it is not required that ALL parameter object instances have a corresponding “ParamN” entry in an EDS. However, when a parameter object instance exists within a node, and if this parameter is also described within an EDS, then the value of “N” in “ParamN” shall be equal to the parameter object instance.

Each entry shall contain the formatted fields shown in Table C.13. The “ProxyParam”and “ProxiedParam” keywords are defined further in C.4.1.5.3.1, as part of the modular EDS requirements.

Table C.13 — Parameter format

Field name	Field number	Data type	Required/Optional
Reserved	1	USINT	Required
Link Path Size	2	USINT	Optional
Link Path	3	EPATH	Optional
Descriptor	4	WORD	Required
Data Type	5	USINT/EPATH	Required
Data Size	6	USINT	Required
Parameter Name	7	EDS_Char_Array	Required
Units String	8	EDS_Char_Array	Required
Help String	9	EDS_Char_Array	Required
Minimum Value	10	data type	Conditional[a]
Maximum Value	11	data type	Conditional[a]
Default Value	12	data type	Required
Scaling Multiplier	13	UINT	Optional
Scaling Divider	14	UINT	Optional
Scaling Base	15	UINT	Optional
Scaling Offset	16	INT	Optional
Multiplier Link	17	UINT	Optional
Divisor Link	18	UINT	Optional
Base Link	19	UINT	Optional
Offset Link	20	UINT	Optional
Decimal Precision	21	USINT	Optional
International Parameter Name	22	STRINGI	Optional
International Engineering Units	23	STRINGI	Optional
International Help String	24	STRINGI	Optional
[a]These are further specified in Table C.17			

The entries in the parameter section shall provide the information as shown in Table C.14 and Table C.18.

Parameter fields listed in Table C.14 are common to all parameters.

Table C.14 — Common parameter fields

Fields	Description
Reserved	This first field shall contain a zero.
Link Path Size	The number of bytes used to represent path. If the link size does not agree with the number bytes in the "Link Path" field, then the "Link Size" shall be ignored. If this parameter is not addressable from the link, this field shall be empty. If this field is empty and the "Link Path" field is not, the "Link Size" shall be equal to the number of bytes in the "Link Path" field.
Link Path	CIP path to the object attribute from where the parameter value is retrieved. The path shall be entered as a character array, using the path notation described in IEC 61158-6:2003 (Type 2) and with the format as specified in C.4.1.3.6. If the parameter described by this ParamN entry is not directly addressable from the network, this field shall be empty. If this field contains a null string, “”, the parameter described by this ParamN entry shall be addressable as the data attribute (instance attribute 1) of the Nth instance of the Parameter object (i.e. using path “20 0F 24 N 30 01")
Descriptor	The parameter descriptor. Contains bit flags that describe the behaviour of the individual parameters (see Table C.15)
Data Type	The data type identifier, as defined in IEC 61158-6:2003 (Type 2: Data Type Reporting). This identifier shall be encoded either as a USINT, or an EPATH. NOTE Old versions of EDS files may use USINT data type identifiers as specified in Table C.16, but these are now obsolete. They are provided here for compatibility reasons.
Data Size	The numeric data size value. For string and EPATH data types, this field specifies the number of bytes per character or entry. Therefore, for the STRING and EPATH data types, this value shall be specified as 1. For the STRING2 data type, this shall be specified as a 2. For the STRINGN data type, this shall be specified as the value of “N”.
Parameter Name	The textual parameter name. If necessary, truncation of the retrieved text shall occur to meet the maximum character array length allowed.
Units String	The textual display units character array. If necessary, truncation of the retrieved text shall occur to meet the maximum character array length allowed.
Help String	The textual help character array. If necessary, truncation of the retrieved text occurs to meet the maximum character array length allowed.
Minimum Value	See Table C.17 for meaning and requirement based on parameter data type.
Maximum Value	See Table C.17 for meaning and requirement based on parameter data type.
Default Value	The default numeric value assigned to the parameter data value.
International Parameter Name	The parameter name expressed in STRINGI notation.
International Engineering Name	The engineering units expressed in STRINGI notation.
International Help String	The help string expressed in STRINGI notation.

The bits of the Descriptor field shall be as defined in Table C.15.

Table C.15 — Bit definitions of descriptor field

Bit	Definition	Bit value and meaning
0	Supports settable path	0 = Link path cannot be set. 1 = Link path can be set.
1	Supports enumerated strings	0 = Enumerated strings are not supported.µ 1 = Enumerated strings are supported and may be read
2	Supports scaling	0 = Scaling not supported. 1 = Scaling is supported. The scaling attributes are implemented and the value presented to the user in engineering units.
3	Supports scaling Links	0 = Scaling links not supported. 1 = The values for the scaling attributes may be retrieved from other parameters
4	Read only parameter	0 = Parameter value can be written (set) and read (get). 1 = The parameter value can only be read (get), and not set.
5	Monitor parameter	0 = Parameter value is not updated in real time by the device. 1 = The parameter value is updated in real time by the device.
6	Supports extended precision scaling	0= Extended precision scaling not supported. 1 = Extended precision scaling should be implemented and the value presented to the user in engineering units.
7	Support non-consecutive enumerated strings	0 = Non-consecutive enumerated strings not supported 1 = Non-consecutive enumerated strings are supported
8	Allow both enumeration and individual values	0 = Both enumeration and individual values are not supported 1 = Both enumeration and individual values are supported
9-15	Reserved	These bits are reserved and shall be set to 0.

Old versions of EDS files may use data type identifiers as specified in Table C.16.

Table C.16 — Data types identifiers (obsolete)

Data type identifier	Definition	Data type description
1	WORD	16-bit word
2	UINT	16-bit unsigned integer
3	INT	16-bit signed integer
4	BOOL	Boolean
5	SINT	Short integer
6	DINT	Double integer
7	LINT	Long integer
8	USINT	Unsigned short integer
9	UDINT	Unsigned double integer
10	ULINT	Unsigned long integer
11	REAL	Single floating point format (IEEE 754)
12	LREAL	Double floating point format (IEEE 754)
13	ITIME	Duration (short)
14	TIME	Duration
15	FTIME	Duration (high resolution)
16	LTIME	Duration (long)
17	DATE	Date
18	TIME_OF_DAY	Time of day
19	DATE_AND_TIME	Date and time
20	STRING	8-bit per character string
21	STRING2	16-bit per character string
22	STRINGN	N-byte per character string
23	SHORT_STRING	Short N-byte character string
24	BYTE	8-bit string
25	DWORD	32-bit string
26	LWORD	64-bit string

Table C.17 specifies the meaning and specific requirements for the minimum and maximum value entries, based on the parameter data type.

Table C.17 — Semantics for minimum and maximum value entries

Data type	Description and semantics	Minimum value semantics	Maximum value semantics	Required/ Optional/ Not allowed
BYTE	Bit String – 8 bit length	The minimum and maximum values for these data types are not defined and shall not be specified in an EDS file.		Not allowed
WORD	Bit String – 16 bit length			
DWORD	Bit String – 32 bit length			
LWORD	Bit String – 64 bit length			
STRING[a]	String (2 byte length indicator, 1 byte per character)	Minimum string length	Maximum string length	Required
STRING2[a]	String (2 byte length indicator, 2 bytes per character)	Minimum string length	Maximum string length	Required
STRINGN[a]	String (2 byte length indicator, N bytes per character)	Minimum string length	Maximum string length	Required
SHORT_STRING[a]	Character string (1 byte length indicator, 1 byte characters)	Minimum string length	Maximum string length	Required
EPATH[a]	Enumerated Path	Minimum string length	Maximum string length	Optional
All Other Data Types		The minimum numeric value that may be assigned to the data value.	The maximum numeric value that may be assigned to the data value.	Optional[b]

[a]The STRING, STRING2, STRINGN, SHORT_STRING and EPATH data types do not have a minimum or maximum value specification. The minimum and maximum value fields are used to present the minimum and maximum string or path lengths. In these cases, the Data Size parameter is used to represent the number of bytes required per character or encoding entry.

[b]If the Minimum Value and/or Maximum Value is not specified then the minimum and/or maximum value for the parameter data value are as defined in IEC 61158-5:2003 (Type 2), based on the parameter data type.

Parameter fields listed in Table C.18 are optional and are only meaningful when used with the following data types: SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL and LREAL. Specification of these fields with any other data type is prohibited.

Table C.18 — Parameter fields reserved for numeric data types

Fields	Description
Scaling Multiplier	The numeric multiplier value applied to the current parameter data value.
Scaling Divider	The numeric divisor value applied to the current parameter data value.
Scaling Base	The numeric base value applied to the current parameter data value.
Scaling Offset	The numeric offset value applied to the current parameter data value.
Multiplier Link	The parameter number pointing to a Parameter Object instance or other object attribute that contains the numeric multiplier value to apply to the current parameter data value.
Divisor Link	The parameter number pointing to a Parameter Object instance or other object attribute that contains the numeric divisor value to apply to the current parameter data value.
Base Link	The parameter number pointing to a Parameter Object instance or other object attribute that contains the numeric base value to apply to the current parameter data value.
Offset Link	The parameter number pointing to a Parameter Object instance or other object attribute that contains the numeric offset value to apply to the current parameter data value.
Decimal Precision	The numeric precision value applied to the current parameter data value.

Scaling shall not be performed by the device containing the parameter, but by the display tool. If scaling is supported, the display tool shall use the equation shown in Figure C.5 to determine the parameter engineering value (i.e. the value to display) from the parameter actual value. If scaling is not supported then the parameter value shall be displayed as is.

$$EngValue = \frac{(ActualValue + Offset) * Mult * Base}{Div * 10^{DecimalPrecision}}$$

[a] If extended scaling is not supported, this formula shall be used with DecimalPrecision=0

Figure C.5 — Parameter scaling formula

A second keyword may also exist in the [Params] section. This keyword shall be used to provide an enumeration list of parameter choices to present to the user. The entry keyword for all enumerated parameters shall consist of the character array, “Enum”, combined with the decimal number from the corresponding Param entry. Each Enum entry shall consist of pairs of integers and strings.

Figure C.6 is an example that shows a typical Parameter section.

```
[Params]
   Param1 = 0, 1,"20 02", 0x0E94, 1, 1,"Preset","V","User Manual p33",
          0, 5, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2;

   Param2 =                $ parameter instance
   0,                      $ First field shall equal 0
   6, "20 04 24 01 30 03", $ path size, path
   0x0A94,                 $ descriptor - in hex format
   1,                      $ data type
   1,                      $ data size
   "Trigger",              $ name
   "Hz",                   $ units
   "User Manual p49",      $ help string
   0, 2, 0,                $ min, max, default data values
   1, 1, 1, 0,             $ mult, div, base, offset scaling
    ,  ,  ,  ,             $ mult, div, base, offset links not used
   2;                      $ decimal places

   Param3 =                $ not addressable from link
   0, , , 0x0082, 8, 1, "speed control", "", "", 3, 12, 3, ,,,,,,,,;

   Enum3  = 3, "stop", 8, "slow", 12, "fast";
```

Figure C.6 — [Params] section example

C.4.1.4.7 Parameter Groups section

The parameter groups section shall identify all of the parameter groups in a device. Each parameter group shall contain a list of the parameters in the group. The entry keyword for each group shall consist of the combination of the character array, “Group”, and a parameter group number (decimal), for example “Group1”. The decimal numbers shall start at one and increment by one.

The actual Parameter Group object instance may, but need not be, implemented in the device. Conversely, it is not required that ALL Parameter Group object instances have a corresponding “GroupN” entry in an EDS. However, when a Parameter Group object instance exists within a node, and if this Parameter Group is also described within an EDS, then the value of “N” in “GroupN” shall be equal to the Parameter Group object instance.

The fields in each entry shall contain the group name, the number of members in the group, and the instance numbers of the parameters in the group. The parameter group section shall contain the fields shown in Table C.19.

Table C.19 — Parameter group format

Field name	Field number	Data type	Required/Optional
Group Name String	1	EDS_Char_Array	Required
Number of Members	2	UINT	Required
Parameter	3 through (number of members + 2)	UINT	Required

Figure C.7 is an example that shows a typical Parameter Groups section.

```
[Groups]
   Group1 = "Setup", 2, 1, 2;              $ group 1
   Group2 = "Monitor", 2, 2, 3;            $ group 2
   Group3 = "Maintenance", 2, 1, 3;        $ group 3
```

Figure C.7 — [Groups] section example

C.4.1.4.8 Assembly section

The Assembly section describes the structure of a data block. Often this block is the data attribute of an Assembly object; however, this section of the EDS can be used to describe any complex structure. The description of this data block parallels the mechanism that the Assembly object uses to describe its member list.

The "Revision" entry keyword shall have one 16-bit integer field that shall be the revision (class attribute 1) of the Assembly object within the device. If this optional entry is missing, the revision of the Assembly object shall be 2.

The entry keyword for all assemblies shall consist of one of the following character arrays, “Assem”, “ProxyAssem”, “ProxiedAssem”combined with the Assembly object instance number (decimal) for the device, e.g. “Assem1”. If a particular instance of the Assembly object is addressable from the link, there shall be a one-for-one pairing between the Assem number in the EDS file and the Assembly instance number in the device. The “ProxyAssem”and “ProxiedAssem” keywords are defined further in C.4.1.5.3.2, as part of the modular EDS requirements.

Each entry shall contain the formatted fields shown in Table C.20.

Table C.20 — AssemN keyword format

Field name	Field number	Data type	Required/ Optional
Name	1	EDS_Char_Array	Optional
Path	2	EDS_Char_Array	Optional
Size	3	UINT	Conditional
Descriptor	4	WORD	Optional
Reserved	5, 6	empty	
Member Size	7, 9, 11 …	UINT	Conditional
Member Reference	8, 10, 12 …	AssemN, ProxyAssemN, ParamN, ProxyParamN, UDINT, or EPATH	Conditional

The first field, called "Name", shall be a string giving a name to the data block. This optional field may be used by a user interface.

The second field, called "Path", shall be a string that specifies a logical path. This path shall identify the address of the data block within the device. If the block described by this AssemN entry is not directly addressable from the link, this field shall be empty. If this field is a null string, “”, the data block shall be addressable as the data attribute (instance attribute 3) of the Nth instance of the Assembly object.

The third field, called "Size", shall be the size of the data block in bytes. If neither this field nor the "Member Size"/"Member Reference" fields are present the size of the data block shall be 0. Both of these fields may be present; however, since they both specify the size of the block, the sizes specified by both means shall agree.

The fourth field, “Descriptor”, shall be a bit-field that describes certain properties of the Assembly. The bits of this field shall be interpreted as specified in Table C.21.

Table C.21 — Bit definition of Assembly descriptor field

Bit	Name	Meaning
0	Allow Value Edit	If this bit is set (1), the contents of the Assembly’s member references fields defined as values may be edited. If reset (0), the contents of these member references fields may not be edited. If this field is empty, the meaning shall default to reset (0). The member references considered to be values are those specifying either a UDINT constant, or a path composed of Data Segments.
1-15		Reserved

Fields 5 and 6 shall be reserved and shall be empty.

The remaining fields shall be paired such that a "Member Size" field is paired with a "Member Reference" field making the total number of fields even. The number of fields pairs in each entry shall be variable. The pairs shall correspond to an Assembly object member list.

The allowed values for the "Member Reference" field shall be one of the following:

— a ParamN or ProxyParamN reference from the [Params] section;

— an AssemN or ProxyAssemN reference from the [Assembly] section;

— a string representing a path (EPATH);

— a UDINT constant;

— an empty field;

— additional values as defined for modular EDS in C.4.1.5.3.2.

If the "Member Reference" field is empty, the number of bits specified by the "Member Size" field shall be used as a pad within the Assembly object. A "Member Reference" field containing a null string shall be treated as if the field was empty. A "Member Reference" field and its corresponding "Member Size" shall not both be empty. If the "Member Reference" field specifies an EPATH, this path shall consist of either Logical Segments (path to an object within the device) or Data Segments.

The "Member Size" field shall have units of bits. If a "Member Size" field is empty, the defined size of the corresponding "Member Reference" field shall be used. The defined size of a Param entry shall be as given in its 6th field (size). The defined size of an Assem entry shall be as given in its 3rd field (size).

The members shall be placed into the data block least significant bit first just as they are in the Assembly object. If a "Member Size" field is smaller than the defined size of the corresponding "Member Reference" field, the least significant bits of the corresponding "Member Reference" field shall be used. If a "Member Size" field is larger than the defined size of the corresponding "Member Reference" field, the entire member shall be followed by zero pads to extend the member to the "Member Size". The data block represented shall be an integer number of bytes. The total of all member sizes shall equal the AssemN Size field (when expressed as bits).

Figure C.8 is an example that shows a typical Assembly section. In this example, Assem5 is 1 byte long and has a default value of 0x21.

```
[Params]
   Param1 =
          0,                      $ first field shall equal 0
          6, "20 0F 24 01 30 01", $ path size, path
          0x0000,                 $ descriptor
          2,                      $ data type : 16-bit WORD
          2,                      $ data size in bytes
          "Idle state",           $ name
          "",                     $ units
          "User Manual p48",      $ help string
          0, 2, 1,                $ min, max, default data values
          0, 0, 0, 0,             $ mult, dev, base, offset scaling not used
          0, 0, 0, 0,             $ mult, dev, base, offset link not used
          0;                      $ decimal places not used

   Param2 =
          0, 6, "20 0F 24 02 30 01", $ path size, path
          0x0000, 2, 2,
          "Fault state", "", "User Manual p49",
          0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0;

[Assembly]
   Revision = 2;

   Assem5 = "configuration", "20 04 24 05 30 03",1,,,,
             4, Param1,
             3, Param2,
             1, ;
```

Figure C.8 — [Assembly] section example

NOTE The keyword Variant, combined with a decimal number (e.g. “Variant1”) is reserved for the future definition of new entry types in the Assembly section.

C.4.1.4.9 Connection Manager section

C.4.1.4.9.1 Contents

The Connection Manager section shall contain information concerning the number of types of application connections a device supports. This section is modeled after the Connection Manager Object. Many of the terms used here are described in IEC 61158-5:2003 and IEC 61158-6:2003 (Type 2). Each entry keyword shall be one of the following character arrays, “Connection”, “ProxyConnect”, “ProxiedConnect”, combined with a number (decimal), for example, “Connection1”, “ProxyConnect1”, or “ProxiedConnect1”. The decimal numbers shall start at 1 and increment for each additional “Connection” entry. The decimal number shall not be required to start at 1 or increment for each additional “ProxyConnect” or “ProxiedConnect” entry. The “ProxyConnect”and “ProxiedConnect” keywords are defined further in C.4.1.5.3.3, as part of the modular EDS requirements.

Each entry shall contain the formatted fields shown in Table C.22.

Table C.22 — Connection Manager format

Field name	Field number	Data type	Required/Optional
Trigger and transport	1	DWORD	Required
Connection parameters	2	DWORD	Required
O=>T RPI	3	UDINT or ParamN or ProxyParamN	Optional
O=>T size	4	UINT or ParamN or ProxyParamN	Conditional
O=>T format	5	ParamN or ProxyParamN or AssemN or ProxyAssemN	Conditional
T=>O RPI	6	UDINT or ParamN or ProxyParamN	Optional
T=>O size	7	UINT or ParamN or ProxyParamN	Conditional
T=>O format	8	ParamN or ProxyParamN or AssemN or ProxyAssemN	Conditional
config #1 size	9	UINT or ParamN or ProxyParamN	Optional
config #1 format	10	ParamN or ProxyParamN or AssemN or ProxyAssemN	Optional
config #2 size	11	UINT or ParamN or ProxyParamN	Optional
config #2 format	12	ParamN or ProxyParamN or AssemN or ProxyAssemN	Optional
Connection name string	13	EDS_Char_Array	Required
Help string	14	EDS_Char_Array	Required
Path	15	EDS_Char_Array	Required

C.4.1.4.9.2 Trigger and transport mask

The bit assignments for the trigger and transport mask shall be as shown in Table C.23. A bit shall be set to a 1 (on) for each trigger mode the connection supports. All other bits shall be set to a 0 (off). For the client/server bit: 0=client, 1=server. Only one of the transport types shall be set to a 1 (on).

Table C.23 — Trigger and transport mask bit assignments

Bit	Bit definition
0	class 0: null
1	class 1: duplicate detect
2	class 2: acknowledged
3	class 3: verified
4	class 4: non-blocking
5	class 5: non-blocking, fragmenting
6	class 6: multicast, fragmenting
7-15	class: reserved
16	trigger: cyclic
17	trigger: change of state
18	trigger: application
19-23	trigger: reserved
24	transport type : listen-only
25	transport type : input-only
26	transport type : exclusive-owner
27	transport type : redundant-owner
28-30	reserved
31	client = 0 / server = 1

C.4.1.4.9.3 Connection parameters

The bit assignments for the connection type and priority mask shall be as shown in Table C.24. A bit shall be set to a 1 (on) for each connection type and priority the connection supports. All other bits shall be set to a 0 (off).

Table C.24 — Connection parameters bit assignments

Bit	Bit definition
0	O=>T fixed size supported
1	O=>T variable size supported
2	T=>O fixed size supported
3	T=>O variable size supported
4 – 5	O=>T number of bytes per slot in the O=>T real time data packet for adapter rack connections : 0 = 1 byte 1 = 2 bytes 2 = 4 bytes 3 = 8 bytes
6 – 7	T=>O number of bytes per slot in the T=>O real time data packet for adapter rack connections : 0 = 1 byte 1 = 2 bytes 2 = 4 bytes 3 = 8 bytes
8 – 10	O=>T Real time transfer format. 0 = connection is pure data and is modeless 1 = use zero data length packet to indicate idle mode 2 = reserved 3 = heartbeat 4 = 32-bit run/idle header 5 thru 7 are reserved
11	Reserved
12 – 14	T=>O Real time transfer format 0 = connection is pure data and is modeless 1 = use zero data length packet to indicate idle mode 2 = reserved 3 = heartbeat 4 = 32-bit run/idle header 5 thru 7 are reserved
15	reserved
16	O=>T connection type: NULL
17	O=>T connection type: MULTICAST
18	O=>T connection type: POINT2POINT
19	O=>T connection type: reserved
20	T=>O connection type: NULL
21	T=>O connection type: MULTICAST
22	T=>O connection type: POINT2POINT
23	T=>O connection type: reserved
24	O=>T priority: LOW
25	O=>T priority: HIGH
26	O=>T priority: SCHEDULED
27	O=>T priority: reserved
28	T=>O priority: LOW
29	T=>O priority: HIGH
30	T=>O priority: SCHEDULED
31	T=>O priority: reserved

C.4.1.4.9.4 O=>T RPI

The O=>T RPI shall be the number of microseconds of the requested packet interval. The O=>T RPI shall be a UDINT or a Param or ProxyParam entry from the [Params] section that evaluates to a UDINT. If this field is empty, no constraints are placed on the O=>T RPI.

C.4.1.4.9.5 O=>T size

The O=>T size shall be the number of bytes delivered to the target transport. It shall not include the transport sequence count. The O=>T size shall be a UINT or a Param or ProxyParam entry from the [Params] section that evaluates to a UINT. If this field is empty, the defined size of the O=>T format shall be used after the optional run/idle header size is added.

C.4.1.4.9.6 O=>T format

The O=>T format entry shall define the structure of the consumer buffer for this connection. Valid format descriptors shall be identifiers within the EDS file including

— a Param or ProxyParam entry from the [Params] section;

— an Assem or ProxyAssem entry from the [Assembly] section.

This field may be empty indicating that the consuming format is not specified. This field shall not be empty if the O=>T size field is empty. The O=>T format shall not include the 32-bit run/idle header if it is present.

C.4.1.4.9.7 T=>O RPI

The T=>O RPI shall be the number of microseconds of the requested packet interval. The T=>O RPI shall be a UDINT or a Param or ProxyParam entry from the [Params] section that evaluates to a UDINT. If this field is empty, no constraints are placed on the T=>O RPI.

C.4.1.4.9.8 T=>O size

The T=>O size shall be the number of bytes produced by the target transport. It shall not include the transport sequence count. The T=>O size shall be a UINT or a Param or ProxyParam entry from the [Params] section that evaluates to a UINT. If this field is empty, the defined size of the T=>O format shall be used after the optional run/idle header is added.

C.4.1.4.9.9 T=>O format

The T=>O format entry shall define the structure of the producer buffer for this connection. Valid format descriptors shall be identifiers within the EDS file including

— a Param or ProxyParam entry from the [Params] section;

— an Assem or ProxyAssem entry from the [Assembly] section.

This field may be empty indicating that the producing format is not specified. This field shall not be empty if the T=>O size field is empty. The format shall include the status header if it is present.

C.4.1.4.9.10 Configuration

The config #1 size and config #2 size shall specify the size of the optional data segment that is appended to the path in the Forward_Open. The data segment shall be the concatenation of the two buffers described by the config #1 format and config #2 format. The sizes shall be the number of bytes and shall be a UINT or a Param or ProxyParam entry from the [Params] section that evaluates to a UINT. If one of the config size fields is empty, the natural size of the corresponding config format field shall be used.

Valid config format fields shall be identifiers within the EDS file including

— a Param or ProxyParam entry from the [Params] section;

— an Assem or ProxyAssem entry from the [Assembly] section.

The config format fields may be empty indicating that the config format is not specified. If both the config size and config format fields are empty, no data segment shall be appended to the path of the Forward_Open.

C.4.1.4.9.11 Connection name string

A tool may display the connection name string (character array). The connection name string shall be unique among all Connection entries within the EDS.

C.4.1.4.9.12 Help string

A tool may display the textual help character array. If no help string is to be provided a “null” string shall be used where a null string is defined as two double quotations: “” with no characters between the quotation marks.

C.4.1.4.9.13 Path

A path referencing the target object. The path shall be entered as a CIP Path (EPATH), using the padded path notation described in IEC 61158-6:2003 (Type 2) and with the format as specified in C.4.1.3.6. In addition to the format specified in C.4.1.3.6, the path field may also contain the other following references:

— Param or ProxyParam entries from the [Params] section;

— the keyword SLOT;

— the keyword SYMBOL_ANSI;

— the keyword SLOT_MINUS_ONE.

The Param/ProxyParam entries shall evaluate to a USINT, UINT or UDINT. The value of the Param/ProxyParam shall be used in a little endian order for insertion into the path. The Param/ProxyParam references within the path may be enclosed in brackets as shown in Figure C.9. When enclosed in brackets, the value of the Param/ProxyParam shall be local to the path – the same Param/ProxyParam entry may have a different value elsewhere in the EDS. If the Param/ProxyParam is not enclosed in brackets, the value shall be the same everywhere within the EDS.

The keyword SLOT shall always evaluate to a USINT. The values substituted for the SLOT keyword shall correspond to the position of a module in a backplane.

The keyword SLOT_MINUS_ONE shall always evaluate to a USINT. The values substituted for the SLOT_MINUS_ONE keyword shall correspond to the position of a module in the backplane minus 1.

The keyword SYMBOL_ANSI shall evaluate to an extended symbolic segment (see IEC 61158-6:2003 (Type 2)) entered through the user interface. The extended symbol segment shall be an ANSI extended symbol (CIP path type = 0x91). For example, the string "CAB" shall evaluate to the following extended symbol segment (padded): 0x91 0x03 0x43 0x41 0x42 0x00.

C.4.1.4.9.14 Example Connection Manager section (informative)

Figure C.9 is an example that shows a typical Connection Manager section.

```
[Params]
   Param1 =            $ specifies read buffer
      0, , ,           $ no path means not directly accessible
```

```
      0x0004,          $ descriptor : support scaling
      8, 1,            $ USINT, 1 byte
      "Read",          $ name
      "", "",          $ units & help string
      64, 95, 64,      $ min, max, default data values
      1, 1, 1, -63,    $ mult, div, base, offset scaling
      0, 0, 0, 0, 0;   $ mult, div, base, offset link & decimal
                       $(not used)

   Param2 =            $ specifies write buffer
      0, , ,           $ no path means not directly accessible
      0x0004,          $ descriptor : support scaling
      8, 1,            $ USINT, 1 byte
      "Write",         $ name
      "", "",          $ units & help string
      160, 191, 160,   $ min, max, default data values
      1, 1, 1, -159,   $ mult, div, base, offset scaling
      0, 0, 0, 0, 0;   $ mult, div, base, offset link & decimal
                       $(not used)

[Connection Manager]
   Connection1 =
      0x04010002,      $ trigger & transport
                       $  class 1, cyclic, exclusive-owner
      0x44244401,      $ point/multicast & priority & realtime format
                       $  fixed, 32-bit headers, scheduled,
                       $  O=>T point-to-point, T=>O multicast
      , 16, ,          $ O=>T RPI, size, format
      , 12, ,          $ T=>O RPI, size, format
      , ,              $ config part 1 (not used)
      , ,              $ config part 2 (not used)
      "read/write",    $ connection name
      "",              $ Help string
      "20 04 24 01 2C [Param2] 2C [Param1]";
```

Figure C.9 — [Connection Manager] section example

C.4.1.4.10 Port section

The Port section shall describe the CIP routable ports available within a device. Every CIP routable port shall have a corresponding entry in this section. The entry keyword for all ports shall consist of the character array “Port”, combined with a decimal number corresponding to an instance of the port object. For example, Port1 is instance 1 of the Port Object.

NOTE A CIP routable port is a port that is able to exchange CIP messages with another CIP port connected to another CIP link

Each entry shall contain the formatted fields shown in Table C.25.

Table C.25 — Port entry format

Field name	Field number	Data type	Required/Optional
Port Type Name	1	Field Keyword	Required
Port Name	2	EDS_Char_Array	Optional
Port Object	3	EDS_Char_Array	Optional
Port Number	4	UINT	Required
Reserved	5, 6	empty	Not Used
Port Specific	7, 8, …	Port Specific	Port Specific

The first field, called “Port Type Name”, shall be one of the following field keywords:

— ControlNet;

— ControlNet_Redundant;

— TCP (to indicate an EtherNet/IP capable TCP port);

— DeviceNet;

— a vendor-specific field keyword beginning with the device’s Vendor ID and an underscore character (‘65535_’).

The optional “Port Name” field shall be a string giving a name to the port, and may be used by a user interface. The “Port Object” field shall be a path (EPATH) that identifies the network specific link object associated with the port.

The port number 1 shall correspond to the backplane “port”. Devices with a backplane that cannot route CIP messages shall not have a port number 1.

Figure C.10 is an example that shows a typical Port section.

```
[Port]
   Port1 =   DeviceNet,
             “Port A”,        $ name of port
             “20 03 24 01”,   $ instance one of the DeviceNet object
             2;               $ port number 2

   Port2 =   65535_Chassis,
             “Chassis”,       $ name of port
             “20 9A 24 01”,   $ vendor specific backplane object
             1;               $ port number 1
```

Figure C.10 — [Port] section example

C.4.1.5 Modular EDS file requirements

C.4.1.5.1 General

This subclause describes the concept and contents of a Modular EDS and specifies the usage requirements.

C.4.1.5.2 Modular section

C.4.1.5.2.1 Contents

The [Modular] section shall describe a chassis based system. The two types of modular devices shall be:

— chassis;

— module.

C.4.1.5.2.2 Chassis device

A [Modular] section that describes a chassis shall contain a required keyword “DefineSlotsInRack”. The single field on this entry shall be a 16-bit unsigned integer (UINT) indicating the number of slots in the chassis. Even though an electronic key is defined for the chassis, it need not be addressable from the link. The SLOT keyword used in path definitions in the [Connection Manager] section shall range from 0 to the number of slots minus 1.

The keyword “SlotDisplayRule” is optional. The single field on this entry shall be a parameter from the [Params] section (ParamN only) which defines the translation between internal and external slot number.

Figure C.11 is an example that shows an EDS for a chassis device, including a Modular section.

```
[File]
   DescText = "Wonder Chassis EDS file";
   CreateDate = 09-01-1997;
   CreateTime = 17:23:00;
   Revision = 1.1;

[Device]
   VendCode = 65535;
   VendName = "Widget Works, Inc.";
   ProdType = 101;
   ProdTypeStr = "Widget Works Generic";
   ProdCode = 1;
   MajRev = 1;
   MinRev = 1;
   ProdName = "Widget Chassis";
   Catalog = "1234-chassis";

[Params]
   Param1 =
      0,                        $ first field shall equal 0
      ,,                        $ path size,path
      0x0004,                   $ descriptor
      8,                        $ data type: 32-bit Unsigned Long Integer
      1,                        $ data size in bytes
      "Slot Naming Convention", $ name
      "",                       $ units
      "",                       $ help string
      0,4,0,                    $ min,max,default data values
      0,0,0,0,                  $ mult,dev,base,offset scaling
      0,0,0,0,                  $ mult,dev,base,offset link not used
      0;                        $ decimal places not used

   Enum1 =   0,"n/a",1,"0",2,"1",3,"2",4,"3";

[Modular]
   DefineSlotsInRack   = 5;
   SlotDisplayRule     = Param1;
```

Figure C.11 — [Modular] section describing a chassis

C.4.1.5.2.3 Module device (basic entries)

A [Modular] section that describes a module shall contain the "Width" and "Rack" entries.

The required entry with the keyword "Width" shall have a single field that indicates how many slots of the chassis are consumed by the module. The field shall be a 16-bit unsigned integer (UINT).

The entry keyword for all chassises, into which the module can be placed, shall consist of the character array, "Rack", combined with a decimal number. The numbers shall start at 1 for the first chassis, and shall be incremented for each additional chassis. The fields for the "Rack" entries shall be as shown in Table C.26.

Table C.26 — Rack entry format

Field name	Field number	Data type	Required/Optional
Vendor ID	1	UINT	Required
Product Type	2	UINT	Required
Product Code	3	UINT	Required
Major Revision	4	USINT	Required
Minor Revision	5	USINT	Required
reserved	6, 7, 8	empty	not used
Legal Slot	9, 10, 11 ...	UINT	Required

The "Vendor ID", "Product Type", "Product Code", "Major Revision" and "Minor Revision" field shall identify the electronic key of the chassis into which the module may be placed. The reserved field shall be empty. The "Legal Slot" fields shall identify the slots into which the module may be placed. The EDS for the module shall contain one "Rack" entry for each chassis into which the module may be placed.

Figure C.12 is an example that shows a typical [Modular] section.

```
[Modular]
   Width = 1;

   Rack1 =                        $ this module can plug into
      65535, 101, 1, 1, 1,,,,     $ slots 1, 2, 3 and 4 of
      1, 2, 3, 4;                 $ this five slot chassis
```

Figure C.12 — [Modular] section example

C.4.1.5.2.4 Module device (additional entries)

Overview

Additional entries are defined in the EDS to allow device identification and device keying for modules in chassis-based systems which do not support CIP.

For that purpose, modular devices are typically divided into two categories:

— modules that have a CIP link connection, a corresponding Identity object addressable from the link, and are placed in slot 0 (e.g. communication adapters);

— modules that do not have a CIP link connection or an addressable Identity object, and therefore may not be placed in slot 0 (e.g. I/O modules).

NOTE CIP provides other mechanisms for device identification and device keying for modules supporting a CIP link addressable Identity object.

Entries for a module that does not have a link addressable Identity object

A [Modular] section that describes a module that does not have a link addressable Identity object may contain the entry keyword "ExternalID". The keyword shall have a single field. This field shall be a byte string that identifies the module. This byte string shall be encoded using the same format as specified for an EPATH.

Figure C.13 is an example that shows a typical [Modular] section describing a module without a link addressable Identity object.

```
[Modular]
   Width = 1;

   Rack1 =                        $ this module can plug into
      65535, 101, 1, 1, 1,,,,     $ slots 1, 2, 3 and 4 of
      1, 2, 3, 4;                 $ this five slot chassis

   Rack2 =
      65535, 101, 2, 1, 1,,,,
      1, 2, 3, 4, 5, 6, 7;

   ExternalID = “12 34”;
```

Figure C.13 — [Modular] section example (module without a link addressable Identity object)

Entries for a module that has a link connection and is placed in slot 0

A [Modular] section that describes a module that has a link connection and is placed in slot 0 may contain any of the following entry keywords, or a combination of them.

The keyword “GenericID” shall have a single field. This field shall be a byte string that shall be included in the data segment for a module connection in place of the ExternalID when no module keying is desired. This byte string shall be encoded using the same format as specified for an EPATH.

The keyword “ExternIDExactMatch” shall have a single field, with a value of Yes or No. Yes shall indicate that the ExternalID specifies one specific device, No shall indicate that the ExternalID specifies one of a set of compatible devices. If the “ExternIDExactMatch” keyword is omitted, the default condition shall be that the ExternalID specifies one specific device.

The keyword “Query” shall have 4 fields. The first field shall be a path that identifies a link addressable attribute that contains an array of external identifiers, one for each slot in the chassis except slot 0. The second field shall be the service to use with the query path (i.e. 1 – get attribute all or 14 – get attribute single). The third field shall be an integer that determines the number of bytes used to identify each module and shall be in the range 1 to 16. If a double slot module is in the chassis, the external identifier for the module shall appear twice in the array returned from a query. A query shall only be addressed to a module in slot 0. The fourth field shall be the ExternalID returned when an empty slot exists, encoded using the same format as specified for an EPATH.

Figure C.14 is an example that shows a typical [Modular] section describing a module with a link connection placed in slot 0.

```
 [Modular]
   Width = 1;

   Rack1 =                        $ this module can only plug into
      65535, 101, 1, 1, 1,,,,     $ slot 0 of this five slot chassis
```

```
      0;

   Rack2 = 65535, 101, 2, 1, 1,,,,      0;

   Query = “20 04 24 07 30 03”,1,2,”FF FF”;

   GenericID = “00 00”;

   ExternalIDExactMatch = No;
```

Figure C.14 — [Modular] section example (module with a link connection in slot 0)

C.4.1.5.3 Modular additions to basic EDS sections

C.4.1.5.3.1 Additions to the Parameter section

The “ProxyParam” and “ProxiedParam” keywords shall be used to describe parameters that are proxied by a EtherNet/IP adapter device to another device that does not support the CIP protocol. An example of this is a EtherNet/IP adapter module (the device proxying the connection) in a multiple slot I/O rack with an analog I/O module (the device the connection is proxied for).

The “ProxyParam” shall exist in the EDS for the device that performs the proxy.

The “ProxiedParam” keyword shall exist in the EDS for the device that the proxy is performed for.

The information in the [Modular] section shall be used to associate EDS files containing “ProxyParam” keywords to EDS files containing “ProxiedParam” keywords. This association shall exist when both EDS files specify a matching Rack entry.

The decimal number (that is combined with “ProxyParam” and “ProxiedParam”) shall be used to match a “ProxyParam” to a “ProxiedParam”. The field values of a matched “ProxyParam” and “ProxiedParam” pair shall be combined to constitute the same field value information that exists in a single “Param” entry. This combination shall be done by using the field value from the “ProxyParam” unless that field value is the keyword “Module”. When the field value specified in the “ProxyParam” is “Module” the field value specified in the “ProxiedParam” shall be used. It shall be legal to specify field values for “ProxiedParam” entries whose corresponding field value in the “ProxyParam” is not “Module”, however, these field value shall not by used, they shall exist only for documentation.

Another keyword may also exist in the [Params] section. This keyword shall be used to provide minimum, maximum and default values to be added to the “ProxyParam” minimum, maximum and default values. This entry keyword shall be “ProxyParamSizeAdder”, combined with the decimal number from the corresponding “ProxyParam” entry. Each “ProxyParam” entry shall consist of a Minimum Value, Maximum Value and Default Value fields. The definition of these fields matches the “Param” definitions. The “ProxyParamSizeAdder” keyword provides a means for an adapter on a module connection (e.g. “ProxyConnect”) to add adapter data to the module data and return the combined data on the connection.

Another keyword may also exist in the [Param] section that corresponds to the “ProxyParam”, “ProxyEnum”. “ProxyEnum” has the same definition as “Enum” except it is associated with “ProxyParam” instead of “Param”. A second keyword may also exist in the [Param] section that corresponds to the “ProxiedParam”, “ProxiedEnum”. “ProxiedEnum” has the same definition as “Enum” except it is associated with “ProxiedParam” instead of “Param”.

C.4.1.5.3.2 Additions to the Assembly section

Additional entry keywords

The “ProxyAssem” and “ProxiedAssem” keywords shall be used to describe assemblies that are proxied by a CIP adapter device to another device that does not support the CIP protocol. An example of this is an EtherNet/IP adapter module (the device proxying the connection) in a multiple slot I/O rack with an analog I/O module (the device the connection is proxied for).

The “ProxyAssem” keyword shall exist in the EDS for the device that performs the proxy; the “ProxiedAssem” keyword shall exist in the EDS for the device that the proxy is performed for.

The information in the [Modular] section shall be used to associate EDS files containing “ProxyAssem” keywords to EDS files containing “ProxiedAssem” keywords. This association shall exist when both EDS files specify a matching Rack entry.

The decimal number (that is combined with “ProxyAssem” and “ProxiedAssem”) shall be used to match a “ProxyAssem” to a “ProxiedAssem”. The field values of a matched “ProxyAssem” and “ProxiedAssem” pair shall be combined to constitute the same field value information that exists in a single “Assem” entry. This combination shall be done by using the field value from the “ProxyAssem” unless that field value is one of the keywords “Module” or “ModuleMemberList”. When the field value specified in the “ProxyAssem” is “Module” the field value specified in the “ProxiedAssem” shall be used. The field value “Module” shall not be used for “Member Size” or “Member Reference” fields. “ModuleMemberList” shall only be used in place of a “Member Size” and “Member Reference” field pair. When the field value specified in the “ProxyAssem” is “ModuleMemberList” all “Member Size” and “Member Reference” fields specified in the “ProxiedAssem” shall be used. It shall be legal to specify field values for “ProxiedAssem” entries whose corresponding field value in the “ProxyAssem” is not “Module”, however, these field values shall not by used, they shall exist only for documentation.

Additional field keywords

An adapter rack connection is a connection to a rack based adapter device that includes data from modules in the rack. Such a connection may also be used to send configuration and keying data for modules in the rack (e.g. at connection establishment).

The following keywords are additional values allowed for the "Member Reference" field within the Assembly section, that indicate the special purpose intended for the use of the data defined by an assembly member:

— ExternalID;

— InputSlotMask0 or InputSlotMask1;

— OutputSlotMask0 or OutputSlotMask1;

— ConfigSlotMask0 or ConfigSlotMask1.

The “ExternalID” keyword specifies that this assembly member shall contain either a module device “ExternalID” value if device keying is desired, or the “GenericID” value defined in the adapter EDS if module keying is not desired.

The “ExternalID” keyword combined with a decimal number (e.g. ExternalID2) shall be used to allow individual device keying for adapter rack connections. The decimal (positive) number N in “ExternalIDN” specifies slot N in the rack. The “ExternalIDN” keyword specifies that this assembly member shall contain either a module device “ExternalID” value for slot N if device keying is desired for this slot, or the “GenericID” value defined in the adapter EDS if module keying is not desired for this slot.

NOTE Keying is not available for slot 0.

The “InputSlotMask0” or “InputSlotMask1” keyword shall indicate the location of the input slot mask in the assembly. An input slot mask is an array of bits which represents inclusion or exclusion of a module’s target to originator data in an adapter rack connection. If “InputSlotMask0” keyword is used, bit 0 in this array represents slot 0, bit 1 represents slot 1, and so on. If “InputSlotMask1” keyword is used, bit 0 in this array represents slot 1, bit 1 represents slot 2, and so on. “InputSlotMask0” and “InputSlotMask1” shall not be used both in the same assembly. The preceding “Member size” field shall be required.

The “OutputSlotMask0” or “OutputSlotMask1” keyword shall indicate the location of the output slot mask in the assembly. An output slot mask is an array of bits which represents inclusion or exclusion of a module’s originator to target data in an adapter rack connection. If “OutputSlotMask0” keyword is used, bit 0 in this array represents slot 0, bit 1 represents slot 1, and so on. If “OutputSlotMask1” keyword is used, bit 0 in this array represents slot 1, bit 1

represents slot 2, and so on. “OutputSlotMask0” and “OutputSlotMask1” shall not be used both in the same assembly. The preceding “Member size” field shall be required.

The “ConfigSlotMask0” or “ConfigSlotMask1” keyword shall indicate the location of the configuration slot mask in the assembly. A configuration slot mask is an array of bits which represents inclusion or exclusion of a module’s configuration data in the establishement service of an adapter rack connection. If “ConfigSlotMask0” keyword is used, bit 0 in this array represents slot 0, bit 1 represents slot 1, and so on. If “ConfigSlotMask1” keyword is used, bit 0 in this array represents slot 1, bit 1 represents slot 2, and so on. “ConfigSlotMask0” and “ConfigSlotMask1” shall not be used both in the same assembly. The preceding “Member size” field shall be required.

C.4.1.5.3.3 Additions to the Connection Manager section

The “ProxyConnect” and “ProxiedConnect” keywords shall be used to describe connections that are proxied by a CIP adapter device to another device that does not support the CIP protocol. An example of this is a EtherNet/IP adapter module (the device proxying the connection) in a multiple slot I/O rack with an analog I/O module (the device the connection is proxied for).

The “ProxyConnect” keyword entry shall exist in the EDS for the device that performs the proxy. In the example above, this would be the EtherNet/IP adapter module.

The “ProxiedConnect” keyword entry shall exist in the EDS for the device that the proxy is performed for. In the example above, this would be the analog I/O module.

The information in the [Modular] section shall be used to associate EDS files containing “ProxyConnect” keywords to EDS files containing “ProxiedConnect” keywords. This association shall exist when both EDS files specify a matching Rack entry.

The decimal number (that is combined with “ProxyConnect” and “ProxiedConnect”) shall be used to match a “ProxyConnect” to a “ProxiedConnect”. The field values of a matched “ProxyConnect” and “ProxiedConnect” pair shall be combined to constitute the same field value information that exists in a single “Connection” entry. This combination shall be done by using the field values from the “ProxyConnect” except for those fields where the value is the keyword “Module”. In those cases, the field value specified in the associated “ProxiedConnect” shall be used. It shall be legal to specify field values for “ProxiedConnect” entries whose corresponding field value in the “ProxyConnect” entry is not “Module”, however, these field values shall not be used, they shall exist only for documentation. The field value for the “ProxyConnect” “connection name string” field shall not be “Module”, the “ProxyConnect” shall always specify the “connection name string”.

C.4.1.5.3.4 Examples of extended EDS section (informative)

Figure C.15 and Figure C.16 are examples that show the usage of the modular EDS extensions for the Parameter, Assembly and Connection Manager sections.

```
[Params]
   Param1 =         0,,,0x0010,2,2,” Target Error Codes",
                          "","",0,0xFFFF,0,0,0,0,0,0,0,0,0,0;
   ProxyParam1 =    0,,,0x0000,2,2,"input size",
                    "","",Module,Module,Module,0,0,0,0,,,,,0;
   ProxyParamSizeAdder1 = 4,4,4;

[Assembly]
   Assem1 = "connection input format",,,,,,
      32,Param1,
      ,ProxyAssem1,
      ,ProxyAssem2;
   ProxyAssem1 = "real time input format","20 7D 24 SLOT 30 0A",,,,,
      ModuleMemberList;
   ProxyAssem2 = "real time status format","20 7D 24 SLOT 30 0B",,,,,
      ModuleMemberList;
```

```
[Connection Manager]
   ProxyConnect1 = 0x010100002, 0x44244401,
          2, 0, ,  2, ProxyParam1, Assem1, , , , , "Listen Only", "",
          "01 SLOT_MINUS_ONE 20 04 24 03 2C 04 2C 02";
```

Figure C.15 — Example of ProxyParam and ProxyAssem entries

```
[Params]
   ProxiedParam1 = ,,,,,,"input size","","",0,2,2,,,,,,,,;

[Assembly]
   ProxiedAssem1 = "real time input format",,,,,;
   ProxiedAssem2 = "real time status format",,,,,,16,;

[Connection Manager]
   ProxiedConnect1 = ,,,0,,,,,,,,,,;
```

Figure C.16 — Example of matching ProxiedParam and ProxiedAssem entries

C.4.2 EtherNet/IP specific EDS requirements

C.4.2.1 EtherNet/IP EDS content

This subclause specifies the file encoding requirements of the Electronic Data Sheet (EDS) which are specific for EtherNet/IP-based networks.

Table C.27 summarizes the structure of the sections which may be present in an EtherNet/IP EDS, the corresponding legal section delimiters, and the order of these sections in an EDS. Some of these sections are common to several CIP-based networks, their specific use for EtherNet/IP is further specified in C.4.2.2 if needed. Other sections may be specific to EtherNet/IP and will be specified in C.4.2.4.

Table C.27 — EtherNet/IP EDS file structure

Section name	Legal delimiter	Placement	Required/Optional
File Description	[File]	1	Required
Device Description	[Device]	2	Required
Device Classification	[Device Classification]	[a]	Required
Parameter Class	[ParamClass]	[a]	Optional
Parameters	[Params]	[a]	Optional
Parameter Groups	[Groups]	[a]	Optional
Assembly	[Assembly]	[a]	Optional
Connection Characteristics	[Connection Manager]	[a]	Conditional[b]
Port	[Port]	[a]	Optional
Modular	[Modular]	[a]	Optional
Vendor Specific	[VendorID_vendorspecifickeyword]	Last	Optional

[a] Placement of these groups only needs to follow the Device Description and Device Classification sections

[b] These sections are required if the corresponding functionality is implemented, else they may be omitted

The EtherNet/IP EDS contents shall be further organised as follows:

— all EtherNet/IP EDS files shall contain the Device Classification section, which shall use the legal delimiter [Device Classification], and may be placed anywhere after the File Description section;

— the optional and conditional sections described in this specification may be present in any order provided that no forward references exist within the EDS file.

C.4.2.2 Implementation of common CIP requirements

C.4.2.2.1 Device Classification section

For any EtherNet/IP compliant device, the Device Classification section of its EDS shall contain at least one ClassN keyword entry with its first field set to EtherNetIP. Further sub-classification of the EtherNetIP classification shall be reserved.

C.4.2.2.2 Port section

In the Port section of the EDS, the PortN entry corresponding to an EtherNet/IP compliant port shall be set as follows:

— the “Port Type Name” field shall have a value of “TCP”;

— the optional “Port Object” field shall be set to the path of the TCP/IP Interface object for this port;

— no additional requirements, beyond those specified in the CIP common subclause (see C.4.1.4.10), are placed on the “Port Name” and “Port Number” fields.

NOTE An EDS for an EtherNet/IP device does not directly refer to the link object for the EtherNet/IP port (e.g. the Ethernet Link object) since it can be referenced through the TCP/IP Interface object for the port.

C.4.2.3 Additional data encoding requirements

There are no additional data encoding requirements for EtherNet/IP EDS files.

C.4.2.4 Additional file requirements

There are no additional file requirements for EtherNet/IP EDS files.

Bibliography

[1] ISO/TR 13283:1998, *Industrial automation — Time-critical communications architectures — User requirements and network management for time-critical communications systems*

[2] IEC/PAS 61499-1:2000, *Function blocks for industrial-process measurement and control systems — Part 1: Architecture*

[3] IEC TS 61915:2003, *Low-voltage switchgear and controlgear — Principles for the development of device profiles for networked industrial devices*

[4] MSTC/JOP 1101:1999, *Specification Autonomous Decentralized Protocol R3.0*

[5] ODVA/CI EtherNet/IP:2001, *EtherNet/IP Specification (Release 1.0)*

ICS 25.040.40

Price based on 125 pages