

PRE-RELEASE VERSION (FDIS)



**Programmable controllers –
Part 10: PLC open XML exchange format**

**Automates programmables –
Partie 10: Format d'échange XML ouvert PLC**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.240.30; 35.240.50

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**



FINAL DRAFT INTERNATIONAL STANDARD (FDIS)

PROJECT NUMBER:
IEC 61131-10 ED1

DATE OF CIRCULATION:
2019-02-01

CLOSING DATE FOR VOTING:
2019-03-15

SUPERSEDES DOCUMENTS:
65B/1108/CDV,65B/1137A/RVC

IEC SC 65B : MEASUREMENT AND CONTROL DEVICES	
SECRETARIAT: United States of America	SECRETARY: Mr Angus Low
OF INTEREST TO THE FOLLOWING COMMITTEES: SC 65E	HORIZONTAL STANDARD: <input type="checkbox"/>
FUNCTIONS CONCERNED: <input type="checkbox"/> EMC <input type="checkbox"/> ENVIRONMENT <input type="checkbox"/> QUALITY ASSURANCE <input type="checkbox"/> SAFETY	
<input checked="" type="checkbox"/> SUBMITTED FOR CENELEC PARALLEL VOTING Attention IEC-CENELEC parallel voting The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Final Draft International Standard (FDIS) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	<input type="checkbox"/> NOT SUBMITTED FOR CENELEC PARALLEL VOTING

This document is a draft distributed for approval. It may not be referred to as an International Standard until published as such.

In addition to their evaluation as being acceptable for industrial, technological, commercial and user purposes, Final Draft International Standards may on occasion have to be considered in the light of their potential to become standards to which reference may be made in national regulations.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

TITLE:

Programmable controllers – Part 10: PLC open XML exchange format

PROPOSED STABILITY DATE: 2021

NOTE FROM TC/SC OFFICERS:

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
1.1 General.....	11
1.2 Implementation specific parameters.....	12
2 Normative references.....	13
3 Terms, definitions, abbreviated terms and acronyms.....	13
3.1 General terms and definitions.....	13
3.2 Abbreviated terms.....	13
4 Overview of schema concepts.....	14
4.1 Schema versioning.....	14
4.2 Naming conventions.....	14
4.3 Coordinate system of graphical languages.....	14
4.4 Schema extension concepts.....	18
5 Compliance.....	19
5.1 General.....	19
5.2 Feature tables.....	19
5.3 Vendor's compliance statement.....	19
6 Main schema element "Project".....	20
6.1 General.....	20
6.2 "FileHeader".....	20
6.3 "ContentHeader".....	21
6.4 "Types".....	22
6.5 "Instances".....	22
6.5.1 General ("Configuration").....	22
6.5.2 "Resource".....	23
6.5.3 "AccessVars".....	26
6.5.4 "ConfigVars".....	26
7 Abstract complex types.....	27
7.1 Purpose of abstract complex types.....	27
7.2 Abstract complex types for data type specifications.....	28
7.2.1 General.....	28
7.2.2 "TypeSpecBase".....	28
7.2.3 "InstantlyDefinableTypeSpecBase".....	28
7.3 Abstract complex types for behaviour representations.....	28
7.3.1 General.....	28
7.3.2 "BehaviourRepresentationBase".....	29
7.3.3 "ProgrammingLanguageBase".....	29
7.4 Abstract complex types for graphical objects.....	29
7.4.1 General.....	29
7.4.2 "IdentifiedObjectBase".....	31
7.4.3 "GraphicalObjectBase".....	31
7.4.4 "CommonObjectBase".....	32
7.4.5 "FbdObjectBase".....	32
7.4.6 "LdObjectBase".....	32

7.4.7	"SfcObjectBase"	33
7.4.8	"NetworkBase"	33
7.5	Abstract complex types for textual constructs	34
7.5.1	General	34
7.5.2	"TextualObjectBase"	35
7.5.3	"NamespaceContentBase"	36
7.5.4	"TaskBase"	37
8	Namespace declaration	37
9	User-defined data type declaration	38
9.1	"UserDefinedTypeDecl"	38
9.2	"ArrayTypeSpec"	38
9.3	"DirectlyDerivedTypeSpec"	39
9.4	"EnumTypeSpec"	39
9.5	"EnumTypeWithNamedValueSpec"	40
9.6	"StructTypeSpec"	40
9.7	"SubrangeTypeSpec"	41
9.8	"ReferenceTypeSpec"	41
9.9	"ElementaryType"	42
10	POU declaration	42
10.1	"PouDecl"	42
10.2	"Program"	42
10.3	"FunctionBlock"	44
10.4	"Class"	45
10.5	"Function"	46
10.6	"Interface"	47
10.7	"Action"	47
10.8	"NamedTransition"	48
10.9	"MethodPrototype"	48
10.10	"Method"	49
10.11	"ParameterSet"	51
10.12	"VarListWithAccessSpec"	53
10.13	"AccessSpecifiers"	53
10.14	"Body"	53
10.15	"BodyWithoutSFC"	54
10.16	"Predicate"	54
11	Variable declaration	55
11.1	"VarList"	55
11.2	"ExternalVarList"	56
11.3	"VariableDecl"	56
11.4	"VariableDeclPlain"	57
11.5	"TypeRef"	57
11.6	"Value"	57
11.7	"AddressExpression"	58
11.8	"FixedAddressExpression"	59
12	Behaviour representation	59
12.1	"IL"	59
12.2	"ST"	59
12.3	"FBD"	60

12.4	"FbdNetwork"	60
12.5	"LD"	60
12.6	"LadderRung"	61
12.7	"SFC".....	61
13	Graphical behaviour representation	61
13.1	General.....	61
13.2	Common elements	62
13.2.1	"Comment"	62
13.2.2	"Connector"	62
13.2.3	"Continuation".....	63
13.2.4	"ActionBlocks"	63
13.3	FBD elements	65
13.3.1	"Block".....	65
13.3.2	"graphicalFormalParameterCommon"	68
13.3.3	"DataSource"	68
13.3.4	"DataSink"	69
13.3.5	"Unconnected".....	69
13.3.6	"Jump".....	70
13.3.7	"Return"	71
13.4	LD elements	71
13.4.1	"LeftPowerRail"	71
13.4.2	"RightPowerRail"	72
13.4.3	"Coil"	72
13.4.4	"Contact"	73
13.4.5	"CompareContact".....	74
13.5	SFC elements	75
13.5.1	"Step".....	75
13.5.2	"Transition".....	76
13.5.3	"SelectionDivergence"	77
13.5.4	"SelectionConvergence"	78
13.5.5	"SimultaneousDivergence".....	79
13.5.6	"SimultaneousConvergence".....	79
13.6	Connections.....	80
13.6.1	General	80
13.6.2	"ConnectionPointIn".....	80
13.6.3	"Connection".....	81
13.6.4	"FeedbackConnection"	82
13.6.5	"ConnectionPointOut"	82
14	Resource declaration.....	83
14.1	"StandardTask".....	83
14.2	"ParameterAssignment"	83
15	Miscellaneous.....	83
15.1	"XyDecimalValue"	83
15.2	"AddData"	84
15.3	"TextBase".....	84
15.4	"SimpleText"	84
15.5	"EdgeModifierType"	85
Annex A	(normative) Formal XML exchange format schema definition	86

Annex B (informative) Recommended schemata	162
B.1 General.....	162
B.2 Recommended schemata to be used by "AddData"	165
B.3 Recommended schemata to be used by abstract complex type	173
Annex C (informative) Example XML document.....	191
Bibliography.....	277
Figure 1 – Main overview of XML exchange format usage (example)	11
Figure 2 – Mapping coordinate information to the coordinate system	15
Figure 3 – Transforming position using the scaling information	15
Figure 4 – Objects anchor points and object rectangles examples	18
Figure 5 – Main schema element "Project"	20
Figure 6 – Element "FileHeader"	21
Figure 7 – Element "ContentHeader"	21
Figure 8 – Element "Types"	22
Figure 9 – Element "Instances"	23
Figure 10 – Element "Resource"	24
Figure 11 – Element "ProgramInstance"	25
Figure 12 – Element "AccessVars"	26
Figure 13 – Element "ConfigVars"	27
Figure 14 – Extension relationship among complex types for data type specifications	28
Figure 15 – Extension relationship among complex types for behaviour representations	29
Figure 16 – Extension relationship among complex types for graphical objects	31
Figure 17 – Complex type "IdentifiedObjectBase"	31
Figure 18 – Complex type "GraphicalObjectBase"	31
Figure 19 – Complex type "CommonObjectBase"	32
Figure 20 – Complex type "FbdObjectBase"	32
Figure 21 – Complex type "LdObjectBase"	33
Figure 22 – Complex type "SfcObjectBase"	33
Figure 23 – Complex type "NetworkBase"	34
Figure 24 – Extension relationship among complex types for textual objects	35
Figure 25 – Complex type "TextualObjectBase"	36
Figure 26 – Complex type "NamespaceContentBase"	36
Figure 27 – Complex type "TaskBase"	37
Figure 28 – Complex type "NamespaceDecl"	37
Figure 29 – Complex type "UserDefinedTypeDecl"	38
Figure 30 – Complex type "ArrayTypeSpec"	39
Figure 31 – Complex type "DirectlyDerivedTypeSpec"	39
Figure 32 – Complex type "EnumTypeSpec"	40
Figure 33 – Complex type "EnumTypeWithNamedValueSpec"	40
Figure 34 – Complex type "StructTypeSpec"	41
Figure 35 – Complex type "SubrangeTypeSpec"	41
Figure 36 – Complex type "ReferenceTypeSpec"	41

Figure 37 – Complex type "PouDecl"	42
Figure 38 – Complex type "Program"	43
Figure 39 – Complex type "FunctionBlock"	44
Figure 40 – Complex type "Class"	45
Figure 41 – Complex type "Function"	46
Figure 42 – Complex type "Interface"	47
Figure 43 – Complex type "Action"	47
Figure 44 – Complex type "NamedTransition"	48
Figure 45 – Complex type "MethodPrototype"	49
Figure 46 – Complex type "Method"	50
Figure 47 – Complex type "ParameterSet"	52
Figure 48 – Complex type "VarListWithAccessSpec"	53
Figure 49 – Complex type "Body"	54
Figure 50 – Complex type "BodyWithoutSFC"	54
Figure 51 – Complex type "Predicate"	55
Figure 52 – Complex type "VarList"	55
Figure 53 – Complex type "ExternalVarList"	56
Figure 54 – Complex type "VariableDecl"	56
Figure 55 – Complex type "VariableDeclPlain"	57
Figure 56 – Complex type "TypeRef"	57
Figure 57 – Complex type "Value"	58
Figure 58 – Complex type "AddressExpression"	58
Figure 59 – Complex type "FixedAddressExpression"	59
Figure 60 – Complex type "IL"	59
Figure 61 – Complex type "ST"	59
Figure 62 – Complex type "FBD"	60
Figure 63 – Complex type "FbdNetwork"	60
Figure 64 – Complex type "LD"	60
Figure 65 – Complex type "LadderRung"	61
Figure 66 – Complex type "SFC"	61
Figure 67 – Complex type "Comment"	62
Figure 68 – Complex type "Connector"	62
Figure 69 – Complex type "Continuation"	63
Figure 70 – Complex type "ActionBlocks"	64
Figure 71 – Complex type "Block"	67
Figure 72 – Complex type "DataSource"	68
Figure 73 – Complex type "DataSink"	69
Figure 74 – Complex type "Unconnected"	70
Figure 75 – Complex type "Jump"	70
Figure 76 – Complex type "Return"	71
Figure 77 – Complex type "LeftPowerRail"	71
Figure 78 – Complex type "RightPowerRail"	72
Figure 79 – Complex type "Coil"	73

Figure 80 – Complex type "Contact"	74
Figure 81 – Complex type "CompareContact"	75
Figure 82 – Complex type "Step"	76
Figure 83 – Complex type "Transition"	77
Figure 84 – Complex type "SelectionDivergence"	78
Figure 85 – Complex type "SelectionConvergence"	79
Figure 86 – Complex type "SimultaneousDivergence"	79
Figure 87 – Complex type "SimultaneousConvergence"	80
Figure 88 – Complex type "ConnectionPointIn"	81
Figure 89 – Complex type "Connection"	81
Figure 90 – Complex type "FeedbackConnection"	82
Figure 91 – Complex type "ConnectionPointOut"	82
Figure 92 – Complex type "StandardTask"	83
Figure 93 – Complex type "ParameterAssignment"	83
Figure 94 – Complex type "XyDecimalValue"	84
Figure 95 – Complex type "AddData"	84
Figure 96 – Complex type "TextBase"	84
Figure 97 – Complex type "SimpleText"	85
Figure B.1 – Only IEC 61131-3 features	162
Figure B.2 – Vendor specific extensions "AddData"	163
Figure B.3 – Vendor specific extensions (abstract complex type)	164

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PROGRAMMABLE CONTROLLERS –

Part 10: PLC open XML exchange format

FOREWORD

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

International Standard IEC 61131-10 has been prepared by subcommittee 65B: Measurement and control devices, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

FDIS	Result on voting
65B/xxx/FDIS	65B/xxx/RVD

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

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

A list of all parts in the IEC 61131 series, published under the general title *Programmable controllers*, can be found on the IEC website.

This IEC standard includes Code Components i.e. components that are intended to be directly processed by a computer. Such content is any text found between the markers <CODE BEGINS> and <CODE ENDS>, or otherwise is clearly labelled in this standard as a Code Component.

The purchase of this IEC standard carries a copyright license for the purchaser to sell software containing Code Components from this standard to end users either directly or via distributors, subject to IEC software licensing conditions, which can be found at: www.iec.ch/CCv1.

The Code Components included in this IEC standard are also available as an electronic machine-readable file at <http://www.plcopen.org/technical-activities/IEC61131-10/CodeComponents/PLCopenXML.htm>.

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

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

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

INTRODUCTION

The International Standard IEC 61131 describes programmable logic controllers (PLCs).

IEC 61131-3 defines programming languages. Users want standardized programming languages and the ability to exchange a complete program or parts of that program between different development environments, i.e. from an exporting environment to an importing environment.

IEC 61131-3 defines program organization units (POUs). But an entire program also consists of user-defined data types, global and external declarations and other elements besides the POUs. In this document, the term "IEC 61131-3 project" is used. It contains all above-mentioned language elements, required for an exchange, in order to get a consistent program in the importing environment.

The exchange of POUs developed in one of the textual languages, i.e. instruction list (IL) and structured text (ST) or the textual representation of sequential function charts (SFC) is possible, because a syntax description of these languages is part of the IEC 61131-3 standard. The objective of this document is to extend the reuse of programmed solutions both for textual languages and graphical languages, i.e. function block diagram (FBD) and ladder diagram (LD) or the graphical representation of SFCs. Furthermore, the completeness of exchange between the different environments depends on the supported features that are listed in the compliance list defined in IEC 61131-3.

This document defines a solution independent eXtensible Markup Language (XML) based exchange format, to be supported by interfaces of different kinds of software tools. Beside textual and program logic information, it also provides the ability to transfer graphical representation information, e.g. the position and size of function blocks and how they are connected. The design of the 'transferred' parts shall represent the same program logic, however it may be altered in look and feel.

This document's XML exchange format enables a transfer of IEC 61131-3 projects, from an exporting environment to an importing environment, including extensions for layout and formatting.

This document's XML exchange format can not only describe correct IEC 61131-3 POUs, but it can represent a working state of the IEC 61131-3 project. For example, even if the IEC 61131-3 source project is incomplete, for example if it contains compile errors, it can be represented.

Syntactically incorrect IEC 61131-3 projects can be represented. For example, such a project could be an in-between version or a project containing several unconnected FBD blocks.

This document's XML exchange format provides for life cycle management of automation systems, e.g. in case of redesign, maintenance or device replacement. If an IEC 61131-3 project is stored in this standard's XML exchange format, it could be reused independent of a special development environment. And thus, it could be modified and maintained by any other development environment supporting this standard's XML exchange format.

This International Standard was developed using material from PLCopen[®]1. This document extends PLCopen[®] XML, adopts it to the features of IEC 61131-3:2013 and is therefore not compatible with previous versions of PLCopen[®] XML.

¹ PLCopen[®] is the registered trademark of PLCopen. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

PROGRAMMABLE CONTROLLERS – Part 10: PLC open XML exchange format

1 Scope

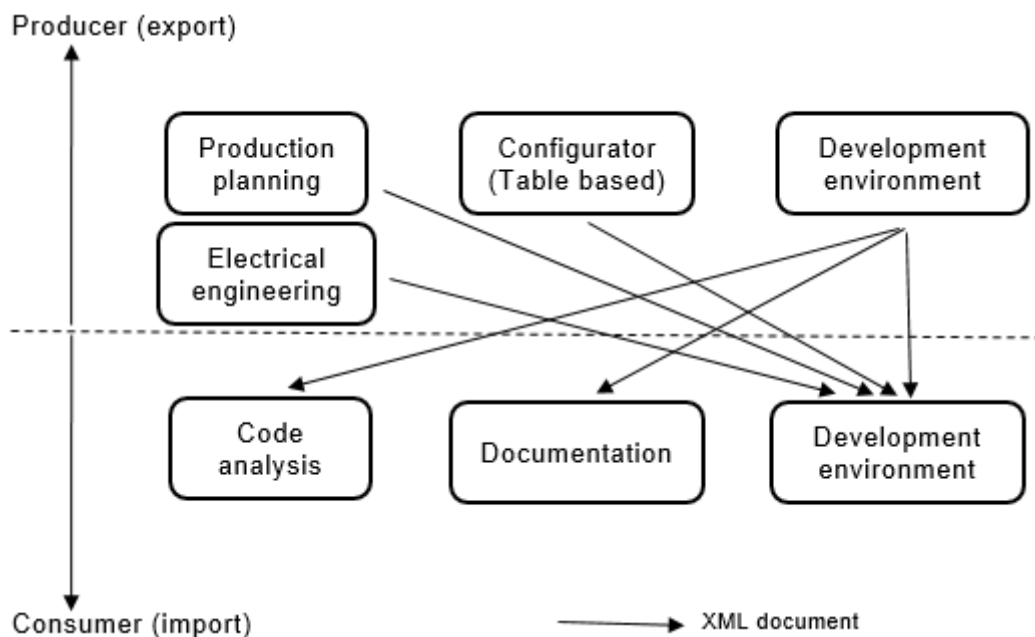
1.1 General

This part of IEC 61131 specifies an XML-based exchange format for the export and import of IEC 61131-3 projects. A complete IEC 61131-3 project implemented in an IEC 61131-3 environment can be transferred between different programming environments. It allows for the exchange of configuration elements, data types, and POUs written in:

- the textual language, instruction list (IL),
- the textual language, structured text (ST),
- the graphical language, ladder diagram (LD),
- the graphical language, function block diagram (FBD), and
- sequential function chart (SFC).

The exchange format is specified as a corresponding XML schema. The XML schema is an independent file with the .xsd extension and as such part of this specification. The specification of this schema is contained in Annex A. Annex B provides recommended schemata for extensions. An example XML document is given in Annex C. It is assumed that the reader of this document is familiar with XML technology.

Figure 1 provides an example overview of the usage of the XML exchange format. Different tools may produce and consume XML based IEC 61131-3 information.



IEC

Figure 1 – Main overview of XML exchange format usage (example)

The usage of the XML exchange format should provide more than a simple export/import from one development environment to another. All relevant information should be exported. This may include coordinate information for graphical tools. The importing tool should be able to filter which parts of this information need to be imported into its destination environment. Vendor-specific information and attributes may be included in the export file and selectively imported, if applicable. The vendor-specific information shall not influence the logic part of the program. Filtering should be done on the import – thus vendors shall ensure that their extensions of the XML schema are done in such a way that neglecting the information during import does not affect the functionality of the IEC 61131-3 project. Vendor specific attributes and information may be added by vendor specific XML schema – besides the XML exchange format defined in this document.

The described formats are designed for the import and export of IEC 61131-3 projects. Such an IEC 61131-3 project can be under development and as a consequence be incomplete.

Concerning the exchange of graphical language constructs between different programming systems, the focus is on logical information with optional explicit graphics.

1.2 Implementation specific parameters

This document does not provide means or requirements for compliant functionality (e.g. functional subset which has to be supported by all Programming and Debugging Tools (PADTs)). This document enables the exchange of all possible features defined in IEC 61131-3. Moreover, many implementation-specific features can be expressed using the AddData mechanism.

In some use cases, programs are either transferred from one PADT to another or generated for the use in a different PADT. In both cases, the function set of these PADTs may be different as well as their settings of implementation-dependent parameters. If several PADTs have to be supported/considered, the functionality of the program has to be restricted to the subset supported by all PADTs in question. Some of these functions can be determined from the IEC 61131-3 feature tables of the concerned PADT, for example:

- supported data types and standard functions,
- pre-emptive or non-pre-emptive scheduling,
- SFC with or without a final scan, etc.

Other functions and settings of implementation dependent parameters may require more effort to determine, for example:

- maximum amounts of code or variables per POU,
- maximum length of identifiers (variable name length),
- size of STRING and WSTRING variables with default length or maximum length,
- SFC to evaluate all transition conditions or only those with active steps as predecessors,
- range and precision of data types TIME, DATE, TOD, DT,
- runtime performance of (the POU in) the PLC,
- execution order within a graphical network, etc.

These differences have to be considered for use cases with more than one PADT. In some cases it may be appropriate to use only functionality supported by all concerned PADTs; in other cases, it may be necessary to manually change and test the program after importing into the PADT.

This document does not state requirements regarding compliant functions of the PADT. It defines an exchange format to exchange programs that are compliant with IEC 61131-3.

2 Normative references

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

IEC 61131-1, *Programmable controllers – Part 1: General information*

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

SOMMAIRE

AVANT-PROPOS	8
INTRODUCTION	10
1 Domaine d'application	12
1.1 Généralités	12
1.2 Mise en œuvre de paramètres spécifiques	13
2 Références normatives	14
3 Termes, définitions, termes abrégés et acronymes	14
3.1 Définitions générales	14
3.2 Termes abrégés	14
4 Aperçu des concepts de schéma	15
4.1 Contrôle des versions de schéma	15
4.2 Conventions d'appellation	15
4.3 Système de coordonnées des langages graphiques	15
4.4 Concepts d'extension de schéma	19
5 Conformité	20
5.1 Généralités	20
5.2 Tableaux des caractéristiques	20
5.3 Déclaration de conformité du fournisseur	20
6 Élément de schéma principal "Project"	21
6.1 Généralités	21
6.2 "FileHeader"	21
6.3 "ContentHeader"	22
6.4 "Types"	23
6.5 "Instances"	23
6.5.1 Généralités ("Configuration")	23
6.5.2 "Resource"	24
6.5.3 "AccessVars"	27
6.5.4 "ConfigVars"	27
7 Types complexes abstraits	28
7.1 Objectif des types complexes abstraits	28
7.2 Types complexes abstraits pour les spécifications de type de données	29
7.2.1 Généralités	29
7.2.2 "TypeSpecBase"	29
7.2.3 "InstantlyDefinableTypeSpecBase"	29
7.3 Types complexes abstraits pour les représentations de comportement	29
7.3.1 Généralités	29
7.3.2 "BehaviourRepresentationBase"	30
7.3.3 "ProgrammingLanguageBase"	30
7.4 Types complexes abstraits pour les objets graphiques	30
7.4.1 Généralités	30
7.4.2 "IdentifiedObjectBase"	32
7.4.3 "GraphicalObjectBase"	32
7.4.4 "CommonObjectBase"	33
7.4.5 "FbdObjectBase"	33
7.4.6 "LdObjectBase"	33

7.4.7	"SfcObjectBase"	34
7.4.8	"NetworkBase"	34
7.5	Types complexes abstraits pour les constructions textuelles	35
7.5.1	Généralités	35
7.5.2	"TextualObjectBase"	36
7.5.3	"NamespaceContentBase"	37
7.5.4	"TaskBase"	38
8	Déclaration d'espace de noms	38
9	Déclaration de type de données défini par l'utilisateur	39
9.1	"UserDefinedTypeDecl"	39
9.2	"ArrayTypeSpec"	39
9.3	"DirectlyDerivedTypeSpec"	40
9.4	"EnumTypeSpec"	40
9.5	"EnumTypeWithNamedValueSpec"	41
9.6	"StructTypeSpec"	41
9.7	"SubrangeTypeSpec"	42
9.8	"ReferenceTypeSpec"	42
9.9	"ElementaryType"	43
10	Déclaration POU	43
10.1	"PouDecl"	43
10.2	"Program"	43
10.3	"FunctionBlock"	45
10.4	"Class"	46
10.5	"Function"	47
10.6	"Interface"	48
10.7	"Action"	48
10.8	"NamedTransition"	49
10.9	"MethodPrototype"	49
10.10	"Method"	50
10.11	"ParameterSet"	52
10.12	"VarListWithAccessSpec"	54
10.13	"AccessSpecifiers"	54
10.14	"Body"	54
10.15	"BodyWithoutSFC"	55
10.16	"Predicate"	55
11	Déclaration de variable	56
11.1	"VarList"	56
11.2	"ExternalVarList"	57
11.3	"VariableDecl"	57
11.4	"VariableDeclPlain"	58
11.5	"TypeRef"	58
11.6	"Value"	58
11.7	"AddressExpression"	59
11.8	"FixedAddressExpression"	60
12	Représentation du comportement	60
12.1	"IL"	60
12.2	"ST"	60
12.3	"FBD"	61

12.4	"FbdNetwork"	61
12.5	"LD"	61
12.6	"LadderRung"	62
12.7	"SFC".....	62
13	Représentation du comportement graphique.....	62
13.1	Généralités	62
13.2	Éléments communs.....	63
13.2.1	"Comment"	63
13.2.2	"Connector"	63
13.2.3	"Continuation".....	64
13.2.4	"ActionBlocks"	64
13.3	Éléments FBD.....	66
13.3.1	"Block".....	66
13.3.2	"graphicalFormalParameterCommon"	69
13.3.3	"DataSource"	69
13.3.4	"DataSink"	70
13.3.5	"Unconnected".....	70
13.3.6	"Jump".....	71
13.3.7	"Return"	72
13.4	Éléments LD	72
13.4.1	"LeftPowerRail"	72
13.4.2	"RightPowerRail"	73
13.4.3	"Coil"	73
13.4.4	"Contact"	74
13.4.5	"CompareContact"	75
13.5	Éléments SFC.....	76
13.5.1	"Step"	76
13.5.2	"Transition".....	77
13.5.3	"SelectionDivergence"	78
13.5.4	"SelectionConvergence"	79
13.5.5	"SimultaneousDivergence".....	80
13.5.6	"SimultaneousConvergence".....	80
13.6	Connexions.....	81
13.6.1	Généralités	81
13.6.2	"ConnectionPointIn".....	81
13.6.3	"Connection".....	82
13.6.4	"FeedbackConnection".....	83
13.6.5	"ConnectionPointOut"	83
14	Déclaration de ressources	84
14.1	"StandardTask"	84
14.2	"ParameterAssignment"	84
15	Divers.....	84
15.1	"XyDecimalValue"	84
15.2	"AddData"	85
15.3	"TextBase".....	85
15.4	"SimpleText"	85
15.5	"EdgeModifierType"	86
Annexe A (normative)	Définition de schéma de format d'échange XML formel	87

Annexe B (informative) Schéma conceptuel recommandé	163
B.1 Généralités	163
B.2 Schéma conceptuel recommandé à utiliser par "AddData"	166
B.3 Schéma conceptuel recommandé à utiliser par le type complexe abstrait.....	174
Annexe C (informative) Exemple de document XML	192
Bibliographie.....	278
Figure 1 – Aperçu général de l'utilisation du format d'échange XML (exemple)	12
Figure 2 – Mapping des informations de coordonnées au système de coordonnées	16
Figure 3 – Transformation de position à l'aide des informations de mise à l'échelle	16
Figure 4 – Exemples de points d'ancrage d'objets et rectangles d'objets.....	19
Figure 5 – Élément de schéma principal "Project"	21
Figure 6 – Élément "FileHeader"	22
Figure 7 – Élément "ContentHeader"	22
Figure 8 – Élément "Types".....	23
Figure 9 – Élément "Instances"	24
Figure 10 – Élément "Resource"	25
Figure 11 – Élément "ProgramInstance".....	26
Figure 12 – Élément "AccessVars"	27
Figure 13 – Élément "ConfigVars"	28
Figure 14 – Relation d'extension entre les types complexes pour les spécifications de type de données	29
Figure 15 – Relation d'extension entre les types complexes pour les représentations de comportement.....	30
Figure 16 – Relation d'extension entre les types complexes pour les objets graphiques.....	32
Figure 17 – Type complexe "IdentifiedObjectBase"	32
Figure 18 – Type complexe "GraphicalObjectBase".....	32
Figure 19 – Type complexe "CommonObjectBase".....	33
Figure 20 – Type complexe "FbdObjectBase"	33
Figure 21 – Type complexe "LdObjectBase".....	34
Figure 22 – Type complexe "SfcObjectBase"	34
Figure 23 – Type complexe "NetworkBase".....	35
Figure 24 – Relation d'extension entre les types complexes pour les objets textuels.....	36
Figure 25 – Type complexe "TextualObjectBase"	37
Figure 26 – Type complexe "NamespaceContentBase"	37
Figure 27 – Type complexe "TaskBase"	38
Figure 28 – Type complexe "NamespaceDecl"	38
Figure 29 – Type complexe "UserDefinedTypeDecl"	39
Figure 30 – Type complexe "ArrayTypeSpec"	40
Figure 31 – Type complexe "DirectlyDerivedTypeSpec"	40
Figure 32 – Type complexe "EnumTypeSpec".....	41
Figure 33 – Type complexe "EnumTypeWithNamedValueSpec"	41
Figure 34 – Type complexe "StructTypeSpec".....	42
Figure 35 – Type complexe "SubrangeTypeSpec".....	42

Figure 36 – Type complexe "ReferenceTypeSpec"	42
Figure 37 – Type complexe "PouDecl"	43
Figure 38 – Type complexe "Program"	44
Figure 39 – Type complexe "FunctionBlock"	45
Figure 40 – Type complexe "Class"	46
Figure 41 – Type complexe "Function"	47
Figure 42 – Type complexe "Interface"	48
Figure 43 – Type complexe "Action"	48
Figure 44 – Type complexe "NamedTransition"	49
Figure 45 – Type complexe "MethodPrototype"	50
Figure 46 – Type complexe "Method"	51
Figure 47 – Type complexe "ParameterSet"	53
Figure 48 – Type complexe "VarListWithAccessSpec"	54
Figure 49 – Type complexe "Body"	55
Figure 50 – Type complexe "BodyWithoutSFC"	55
Figure 51 – Type complexe "Predicate"	56
Figure 52 – Type complexe "VarList"	56
Figure 53 – Type complexe "ExternalVarList"	57
Figure 54 – Type complexe "VariableDecl"	57
Figure 55 – Type complexe "VariableDeclPlain"	58
Figure 56 – Type complexe "TypeRef"	58
Figure 57 – Type complexe "Value"	59
Figure 58 – Type complexe "AddressExpression"	59
Figure 59 – Type complexe "FixedAddressExpression"	60
Figure 60 – Type complexe "IL"	60
Figure 61 – Type complexe "ST"	60
Figure 62 – Type complexe "FBD"	61
Figure 63 – Type complexe "FbdNetwork"	61
Figure 64 – Type complexe "LD"	61
Figure 65 – Type complexe "LadderRung"	62
Figure 66 – Type complexe "SFC"	62
Figure 67 – Type complexe "Comment"	63
Figure 68 – Type complexe "Connector"	63
Figure 69 – Type complexe "Continuation"	64
Figure 70 – Type complexe "ActionBlocks"	65
Figure 71 – Type complexe "Block"	68
Figure 72 – Type complexe "DataSource"	69
Figure 73 – Type complexe "DataSink"	70
Figure 74 – Type complexe "Unconnected"	71
Figure 75 – Type complexe "Jump"	71
Figure 76 – Type complexe "Return"	72
Figure 77 – Type complexe "LeftPowerRail"	72
Figure 78 – Type complexe "RightPowerRail"	73

Figure 79 – Type complexe "Coil"	74
Figure 80 – Type complexe "Contact"	75
Figure 81 – Type complexe "CompareContact"	76
Figure 82 – Type complexe "Step"	77
Figure 83 – Type complexe "Transition"	78
Figure 84 – Type complexe "SelectionDivergence"	79
Figure 85 – Type complexe "SelectionConvergence"	80
Figure 86 – Type complexe "SimultaneousDivergence"	80
Figure 87 – Type complexe "SimultaneousConvergence"	81
Figure 88 – Type complexe "ConnectionPointIn"	82
Figure 89 – Type complexe "Connection"	82
Figure 90 – Type complexe "FeedbackConnection"	83
Figure 91 – Type complexe "ConnectionPointOut"	83
Figure 92 – Type complexe "StandardTask"	84
Figure 93 – Type complexe "ParameterAssignment"	84
Figure 94 – Type complexe "XyDecimalValue"	85
Figure 95 – Type complexe "AddData"	85
Figure 96 – Type complexe "TextBase"	85
Figure 97 – Type complexe "SimpleText"	86
Figure B.1 – Caractéristiques définies dans l'IEC 61131-3 uniquement	163
Figure B.2 – Extensions "AddData" spécifiques au fournisseur	164
Figure B.3 – Extensions spécifiques au fournisseur (type complexe abstrait)	165

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

AUTOMATES PROGRAMMABLES –

Partie 10: Format d'échange XML ouvert PLC

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 61131-10 a été établie par le sous-comité 65B: Équipements de mesure et de contrôle-commande, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Résultat du vote
65B/xxx/FDIS	65B/xxx/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 61131, publiées sous le titre général *Automates programmables*, peut être consultée sur le site web de l'IEC.

La présente norme IEC contient des Composants de code, c'est-à-dire des composants destinés à être traités directement par un ordinateur. Ce type de contenu se présente sous la forme d'un texte placé entre les marqueurs <CODE BEGINS> et <CODE ENDS> ou est clairement étiqueté dans la présente norme en tant que Composant de code.

La présente norme IEC fait l'objet de droits d'auteur relatifs à la vente de logiciels contenant des Composants de code issus de la présente norme aux utilisateurs finaux, soit directement soit par l'intermédiaire de distributeurs, soumis aux termes de la licence logicielle IEC, qui peuvent être consultés à l'adresse suivante: www.iec.ch/CCv1.

Les Composants de code inclus dans la présente norme IEC sont également disponibles sous forme de fichier électronique lisible par une machine, à l'adresse suivante: <http://www.plcopen.org/technical-activities/IEC61131-10/CodeComponents/PLCopenXML.htm>.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "*colour inside*" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La Norme internationale IEC 61131 décrit les automates programmables (PLC – *programmable logic controller*).

L'IEC 61131-3 définit les langages de programmation. Les utilisateurs souhaitent disposer de langages de programmation normalisés et avoir la possibilité d'échanger un programme complet ou des parties de ce programme entre différents environnements de développement, c'est-à-dire d'un environnement d'exportation vers un environnement d'importation.

L'IEC 61131-3 définit des unités d'organisation de programme (POU – *program organization unit*). Toutefois, outre les POU, un programme complet est également composé de types de données définies par l'utilisateur et d'autres éléments. Le présent document utilise le terme "projet IEC 61131-3". Ce terme contient tous les éléments de langages mentionnés ci-dessus, exigés pour assurer l'échange, afin d'obtenir un programme cohérent dans l'environnement d'importation.

Il est possible d'échanger des POU développées dans l'un de ces langages textuels, c'est-à-dire liste d'instructions (IL – *instruction list*) et texte structuré (ST – *structured text*) ou la représentation textuelle du diagramme fonctionnel séquentiel (SFC – *sequential function chart*), une description syntaxique de ces langages faisant partie intégrante de la norme IEC 61131-3. Le présent document a pour objet d'étendre la réutilisation des solutions programmées des langages tant textuels que graphiques, c'est-à-dire le diagramme de bloc fonctionnel (FBD – *function block diagram*) et le diagramme à contacts (LD – *ladder diagram*) ou la représentation graphique du SFC. De plus, l'exhaustivité de l'échange entre les différents environnements dépend des caractéristiques prises en charge qui figurent dans la liste de conformité définie dans l'IEC 61131-3.

Le présent document définit un langage XML (eXtensible Markup Language), solution indépendante reposant sur le format d'échange à prendre en charge par des interfaces de différentes sortes d'outils logiciels. Outre les informations textuelles et de logique de programme, ce langage offre également la possibilité de transférer des informations de représentation graphique (la position et la taille des blocs fonctionnels et la manière dont ils sont reliés, par exemple). La conception des parties "transférées" doit représenter la même logique de programme. Toutefois, son aspect et sa convivialité peuvent être modifiés.

Le format d'échange XML du présent document permet de transférer des projets IEC 61131-3 d'un environnement d'exportation vers un environnement d'importation, y compris les extensions pour la présentation et le formatage.

Le format d'échange XML du présent document peut non seulement décrire les POU correctes de l'IEC 61131-3, mais il peut aussi donner un état d'avancée du projet IEC 61131-3. Par exemple, le projet source IEC 61131-3 peut être représenté même s'il est incomplet (s'il contient des erreurs de compilation, par exemple).

Les projets IEC 61131-3 dont la syntaxe est incorrecte peuvent être représentés. Par exemple, ce type de projet peut être une version transitoire ou un projet contenant plusieurs blocs FBD non reliés.

Le format d'échange XML du présent document assure la gestion du cycle de vie des systèmes d'automatisation (en cas de nouvelle conception, de maintenance ou de remplacement d'un dispositif, par exemple). Si un projet IEC 61131-3 est archivé au format d'échange XML de la présente norme, il peut être réutilisé, quel que soit l'environnement de développement particulier. Ainsi, il peut être modifié et maintenu par un autre environnement de développement prenant en charge le format d'échange XML du présent document.

La présente Norme internationale a été développée à l'aide de matériaux issus de PLCopen®¹. Le présent document étend PLCopen® XML et l'applique aux caractéristiques de l'IEC 61131-3:2013. Il n'est donc pas compatible avec les versions précédentes de PLCopen® XML.

¹ PLCopen® est une marque déposée de PLCopen. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

AUTOMATES PROGRAMMABLES –

Partie 10: Format d'échange XML ouvert PLC

1 Domaine d'application

1.1 Généralités

La présente partie de l'IEC 61131 spécifie un format d'échange XML pour l'exportation et l'importation de projets IEC 61131-3. Un projet IEC 61131-3 complet mis en œuvre dans un environnement IEC 61131-3 peut être transféré entre différents environnements de programmation. Il permet l'échange d'éléments de configuration, de types de données et de POU écrits en :

- langage textuel, liste d'instructions (IL),
- langage textuel, texte structuré (ST),
- langage graphique, diagramme à contacts (LD),
- langage graphique, diagramme de bloc fonctionnel (FBD), et
- diagramme fonctionnel séquentiel (SFC).

Le format d'échange est spécifié sous la forme d'un schéma XML correspondant. Le schéma XML est un fichier indépendant portant l'extension .xsd et, en tant que tel, fait partie de la présente spécification. La spécification de ce schéma est donnée dans l'Annexe A. L'Annexe B donne les schémas conceptuels recommandés pour les extensions. Un exemple de document XML est donné à l'Annexe C. Par hypothèse, le lecteur du présent document maîtrise la technologie XML.

La Figure 1 donne un aperçu de l'utilisation du format d'échange XML. Différents outils peuvent produire et utiliser les informations IEC 61131-3 basées sur XML.

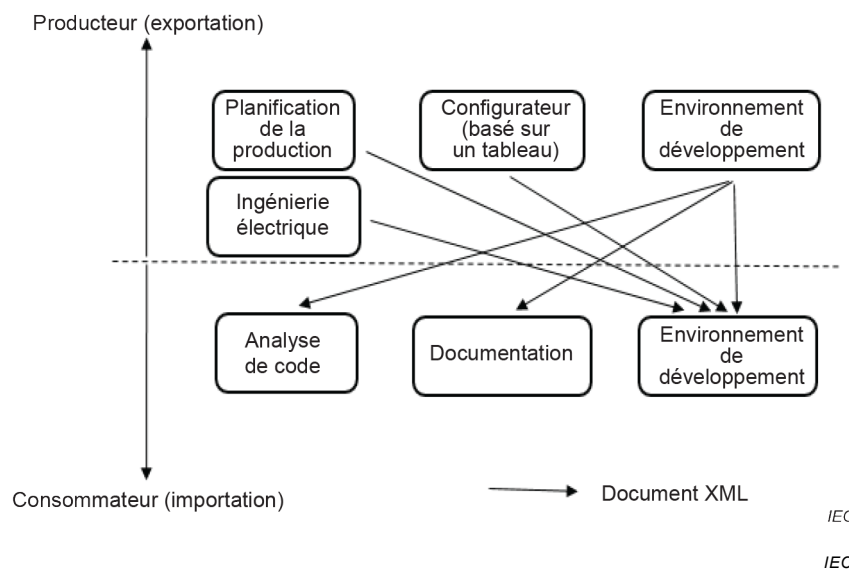


Figure 1 – Aperçu général de l'utilisation du format d'échange XML (exemple)

Il convient que l'utilisation du format d'échange XML offre plus qu'un simple processus d'exportation/importation d'un environnement de développement vers un autre. Il convient d'exporter toutes les informations pertinentes. Cette opération peut inclure les informations de coordonnées pour les outils graphiques. Il convient que l'outil d'importation soit en mesure de filtrer les parties de ces informations qui doivent être importées dans son environnement de destination. Les informations et attributs spécifiques au fournisseur peuvent être inclus dans le fichier d'exportation et importés de manière sélective, le cas échéant. Les informations spécifiques au fournisseur ne doivent pas avoir d'impact sur la partie logique du programme. Il convient de procéder au filtrage lors de l'importation. Les fournisseurs doivent donc vérifier que leurs extensions du schéma XML sont effectuées de sorte que la non-prise en compte d'informations pendant l'importation n'ait aucun impact sur la fonctionnalité du projet IEC 61131-3. Les attributs et informations spécifiques au fournisseur peuvent être ajoutés par le schéma XML spécifique au fournisseur (autre le format d'échange XML défini dans le présent document).

Les formats décrits sont conçus pour l'importation et l'exportation de projets IEC 61131-3. Ce type de projet IEC 61131-3 peut être en cours de développement et, par conséquent, incomplet.

En ce qui concerne l'échange de constructions en langage graphique entre différents systèmes de programmation, les informations logiques contenant des graphiques explicites facultatifs font l'objet de toutes les attentions.

1.2 Mise en œuvre de paramètres spécifiques

Le présent document ne spécifie aucun moyen ni aucune exigence en matière de fonctionnalité conforme (par exemple, le sous-ensemble fonctionnel doit être pris en charge par tous les outils de programmation et de débogage (PADT – *programming and debugging tools*). Le présent document permet d'échanger toutes les caractéristiques possibles définies dans l'IEC 61131-3. De plus, de nombreuses caractéristiques spécifiques à la mise en œuvre peuvent être exprimées à l'aide du mécanisme AddData.

Dans certains cas d'utilisation, les programmes sont soit transférés d'un PADT vers un autre, soit générés pour être utilisés dans un PADT différent. Dans les deux cas, l'ensemble de fonctions de ces PADT peut être différent, de même que les réglages de leurs paramètres dépendant de la mise en œuvre. Si plusieurs PADT doivent être pris en charge/pris en compte, la fonctionnalité du programme doit être limitée au sous-ensemble pris en charge par tous les PADT concernés. Certaines de ces fonctions peuvent être déterminées à partir des tableaux de caractéristiques de l'IEC 61131-3 pour le PADT concerné, par exemple:

- les types de données et fonctions normatives pris en charge,
- la planification préemptive ou non préemptive,
- SFC avec ou sans balayage final, etc.

D'autres fonctions et réglages des paramètres dépendant de la mise en œuvre peuvent exiger de déterminer, par exemple:

- les quantités maximales de codes ou de variables par POU,
- la longueur maximale des identificateurs (longueur de nom de variable),
- la taille des variables STRING et WSTRING avec la longueur par défaut ou la longueur maximale,
- le SFC pour évaluer toutes les conditions de transition ou uniquement celles avec des étapes actives comme prédécesseurs,
- la plage et la précision des types de données TIME, DATE, TOD, DT,
- les performances d'exécution (de la POU dans le) PLC,
- l'ordre d'exécution dans un réseau graphique, etc.

Ces différences doivent être prises en compte pour les cas d'utilisation comportant plusieurs PADT. Dans certains cas, il peut être judicieux de n'utiliser que la fonctionnalité prise en charge par tous les PADT concernés. Dans d'autres cas, il peut s'avérer nécessaire de modifier manuellement et de soumettre à l'essai le programme après importation dans le PADT.

Le présent document ne stipule pas d'exigences relatives aux fonctions conformes du PADT. Il définit un format d'échange de programmes conformes à l'IEC 61131-3.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61131-1, *Automates programmables – Partie 1: Informations générales*

IEC 61131-3, *Automates programmables – Partie 3: Langages de programmation*