AS IEC 61131.10:2026
IEC 61131-10:2019

Programmable controllers

Part 10: PLC open XML exchange format

AS IEC 61131.10:2026

This Australian Standard® was prepared by IT-006, Industrial Process Measurement, Control and Automation. It was approved on behalf of Standards Australia's Standards Development and Accreditation Committee on 10 December 2025.

This Standard was published on 6 February 2026.

The following are represented on Committee IT-006:

Australia Safety Critical Systems Association
Australian Computer Society
Australian Energy Producers
Australian Industry Group
Engineers Australia
Institute of Instrumentation, Control & Automation Australia
ISACA Sydney

This Standard was issued for comment as DR AS IEC 61131.10:2025

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals and new products by visiting:

www.standards.org.au

ISBN: 978 1 76175 611 5

AS IEC 61131.10:2026
IEC 61131-10:2019

Australian Standard®

Programmable controllers

Part 10: PLC open XML exchange format

First published as AS IEC 61131.10:2026.

How to read this Standard

This page explains the meaning of the language and structure of this Standard.

Refer to Standards Australia's Standardisation Guide 006 for more details about drafting rules.

Australian and Australian/New Zealand Standards are voluntary unless they are referenced in legislation or called up in contracts.

Requirements

To conform to a Standard, all requirements in the Standard need to be met.

A requirement is any statement in the Standard which uses the word "shall".

Recommendations, permissions and possibilities

The following words are commonly used in Standards, but statements using them do not have to be followed to conform to the Standard:

(a) "should" means that something is recommended.

(b) "may" means that something is permitted.

(c) "can" means that something is possible.

Structure of Standards

A Standard always has the following parts:

(i) The Preface states who developed the Standard, what the Standard is aiming to do, and how it relates to other documents.

(ii) The Scope states what the Standard is about, what it covers and what it does not cover.

(iii) The Normative references clause lists other documents that are referenced in the Standard as part of requirements.

(iv) The Terms and definitions clause defines important terms to help with understanding the Standard.

A Standard may also include other parts, such as the following:

(1) A normative appendix sets additional requirements that need to be conformed to.

(2) An informative appendix provides additional information or guidance. They usually do not contain requirements. If an informative appendix does contain requirements, the Standard will explain when those requirements apply.

(3) A Bibliography lists documents referenced in the Standard but not as part of requirements.

Many Standards include notes. Notes provide recommendations and/or guidance only. They never contain requirements.

Preface

This Standard was prepared by the Standards Australia Committee IT-006, Industrial Process Measurement, Control and Automation.

The objective of this document is to specify an XML-based exchange format for the export and import of AS IEC 61131.3 projects. A complete AS IEC 61131.3 project implemented in an AS IEC 61131.3 environment can be transferred between different programming environments. It allows for the exchange of configuration elements, data types, and program organization units (POUs) written in:

(a) the textual language, instruction list (IL),

(b) the textual language, structured text (ST),

(c) the graphical language, ladder diagram (LD),

(d) the graphical language, function block diagram (FBD), and

(e) sequential function chart (SFC).

This document is identical with, and has been reproduced from, IEC 61131-10:2019, *Programmable controllers — Part 10: PLC open XML exchange format.*

As this document has been reproduced from an international document, the following apply:

(i) In the source text “this part of IEC 61131” is to be read as “this document”.

(ii) A comma is to be read as a full point when referring to a decimal marker.

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms “normative” and “informative” are used in Standards to define the application of the appendices or annexes to which they apply. A “normative” appendix or annex is an integral part of a Standard, whereas an “informative” appendix or annex is only for information and guidance.

Contents

Page

Preface ..... **v**
FOREWORD ..... **xi**
INTRODUCTION ..... **xiii**
1 Scope **1**
1.1 General 1
1.2 Implementation specific parameters 2
2 Normative references **3**
3 Terms, definitions, abbreviated terms and acronyms **3**
3.1 General terms and definitions 3
3.2 Abbreviated terms 3
4 Overview of schema concepts **4**
4.1 Schema versioning 4
4.2 Naming conventions 4
4.3 Coordinate system of graphical languages 4
4.4 Schema extension concepts 7
5 Compliance **8**
5.1 General 8
5.2 Feature tables 8
5.3 Vendor's compliance statement 8
6 Main schema element "Project" **9**
6.1 General 9
6.2 "FileHeader" 9
6.3 "ContentHeader" 10
6.4 "Types" 11
6.5 "Instances" 11
6.5.1 General ("Configuration") 11
6.5.2 "Resource" 12
6.5.3 "AccessVars" 15
6.5.4 "ConfigVars" 15
7 Abstract complex types **16**
7.1 Purpose of abstract complex types 16
7.2 Abstract complex types for data type specifications 17
7.2.1 General 17
7.2.2 "TypeSpecBase" 17
7.2.3 "InstantlyDefinableTypeSpecBase" 17
7.3 Abstract complex types for behaviour representations 17
7.3.1 General 17
7.3.2 "BehaviourRepresentationBase" 18
7.3.3 "ProgrammingLanguageBase" 18
7.4 Abstract complex types for graphical objects 18
7.4.1 General 18
7.4.2 "IdentifiedObjectBase" 20
7.4.3 "GraphicalObjectBase" 20
7.4.4 "CommonObjectBase" 20
7.4.5 "FbdObjectBase" 21
7.4.6 "LdObjectBase" 21
7.4.7 "SfcObjectBase" 22
7.4.8 "NetworkBase" 22
7.5 Abstract complex types for textual constructs 23
7.5.1 General 23
7.5.2 "TextualObjectBase" 24
7.5.3 "NamespaceContentBase" 25

7.5.4 "TaskBase" ... 26

8 Namespace declaration ... 26

9 User-defined data type declaration ... 27
9.1 "UserDefinedTypeDecl" ... 27
9.2 "ArrayTypeSpec" ... 27
9.3 "DirectlyDerivedTypeSpec" ... 28
9.4 "EnumTypeSpec" ... 28
9.5 "EnumTypeWithNamedValueSpec" ... 28
9.6 "StructTypeSpec" ... 29
9.7 "SubrangeTypeSpec" ... 29
9.8 "ReferenceTypeSpec" ... 30
9.9 "ElementaryType" ... 30

10 POU declaration ... 30
10.1 "PouDecl" ... 30
10.2 "Program" ... 31
10.3 "FunctionBlock" ... 32
10.4 "Class" ... 33
10.5 "Function" ... 34
10.6 "Interface" ... 35
10.7 "Action" ... 35
10.8 "NamedTransition" ... 36
10.9 "MethodPrototype" ... 36
10.10 "Method" ... 37
10.11 "ParameterSet" ... 38
10.12 "VarListWithAccessSpec" ... 40
10.13 "AccessSpecifiers" ... 40
10.14 "Body" ... 40
10.15 "BodyWithoutSFC" ... 41
10.16 "Predicate" ... 41

11 Variable declaration ... 42
11.1 "VarList" ... 42
11.2 "ExternalVarList" ... 43
11.3 "VariableDecl" ... 43
11.4 "VariableDeclPlain" ... 44
11.5 "TypeRef" ... 44
11.6 "Value" ... 44
11.7 "AddressExpression" ... 45
11.8 "FixedAddressExpression" ... 46

12 Behaviour representation ... 46
12.1 "IL" ... 46
12.2 "ST" ... 46
12.3 "FBD" ... 47
12.4 "FbdNetwork" ... 47
12.5 "LD" ... 47
12.6 "LadderRung" ... 48
12.7 "SFC" ... 48

13 Graphical behaviour representation ... 48
13.1 General ... 48
13.2 Common elements ... 49
13.2.1 "Comment" ... 49
13.2.2 "Connector" ... 49
13.2.3 "Continuation" ... 50
13.2.4 "ActionBlocks" ... 50
13.3 FBD elements ... 52
13.3.1 "Block" ... 52
13.3.2 "graphicalFormalParameterCommon" ... 55

13.3.3 "DataSource" ... 55
13.3.4 "DataSink" ... 55
13.3.5 "Unconnected" ... 56
13.3.6 "Jump" ... 57
13.3.7 "Return" ... 57
13.4 LD elements ... 58
13.4.1 "LeftPowerRail" ... 58
13.4.2 "RightPowerRail" ... 58
13.4.3 "Coil" ... 58
13.4.4 "Contact" ... 59
13.4.5 "CompareContact" ... 60
13.5 SFC elements ... 61
13.5.1 "Step" ... 61
13.5.2 "Transition" ... 62
13.5.3 "SelectionDivergence" ... 63
13.5.4 "SelectionConvergence" ... 64
13.5.5 "SimultaneousDivergence" ... 65
13.5.6 "SimultaneousConvergence" ... 65
13.6 Connections ... 66
13.6.1 General ... 66
13.6.2 "ConnectionPointIn" ... 66
13.6.3 "Connection" ... 67
13.6.4 "FeedbackConnection" ... 67
13.6.5 "ConnectionPointOut" ... 68

14 Resource declaration ... 69
14.1 "StandardTask" ... 69
14.2 "ParameterAssignment" ... 69

15 Miscellaneous ... 69
15.1 "XyDecimalValue" ... 69
15.2 "AddData" ... 70
15.3 "TextBase" ... 70
15.4 "SimpleText" ... 70
15.5 "EdgeModifierType" ... 71

Annex A (normative) **Formal XML exchange format schema definition ... 72**

Annex B (informative) **Recommended schemata ... 93**

Annex C (informative) **Example XML document ... 101**

Bibliography ... 125

List of Figures

Page

Figure 1 – Main overview of XML exchange format usage (example) 1
Figure 2 – Mapping coordinate information to the coordinate system 5
Figure 3 – Transforming position using the scaling information 5
Figure 4 – Objects anchor points and object rectangles examples 6
Figure 5 – Main schema element "Project" 9
Figure 6 – Element "FileHeader" 10
Figure 7 – Element "ContentHeader" 10
Figure 8 – Element "Types" 11
Figure 9 – Element "Instances" 12
Figure 10 – Element "Resource" 13
Figure 11 – Element "ProgramInstance" 14
Figure 12 – Element "AccessVars" 15
Figure 13 – Element "ConfigVars" 16
Figure 14 – Extension relationship among complex types for data type specifications 17
Figure 15 – Extension relationship among complex types for behaviour representations 18
Figure 16 – Extension relationship among complex types for graphical objects 19
Figure 17 – Complex type "IdentifiedObjectBase" 20
Figure 18 – Complex type "GraphicalObjectBase" 20
Figure 19 – Complex type "CommonObjectBase" 21
Figure 20 – Complex type "FbdObjectBase" 21
Figure 21 – Complex type "LdObjectBase" 22
Figure 22 – Complex type "SfcObjectBase" 22
Figure 23 – Complex type "NetworkBase" 23
Figure 24 – Extension relationship among complex types for textual objects 24
Figure 25 – Complex type "TextualObjectBase" 25
Figure 26 – Complex type "NamespaceContentBase" 25
Figure 27 – Complex type "TaskBase" 26
Figure 28 – Complex type "NamespaceDecl" 26
Figure 29 – Complex type "UserDefinedTypeDecl" 27
Figure 30 – Complex type "ArrayTypeSpec" 27
Figure 31 – Complex type "DirectlyDerivedTypeSpec" 28
Figure 32 – Complex type "EnumTypeSpec" 28
Figure 33 – Complex type "EnumTypeWithNamedValueSpec" 29
Figure 34 – Complex type "StructTypeSpec" 29
Figure 35 – Complex type "SubrangeTypeSpec" 30
Figure 36 – Complex type "ReferenceTypeSpec" 30
Figure 37 – Complex type "PouDecl" 30
Figure 38 – Complex type "Program" 31
Figure 39 – Complex type "FunctionBlock" 32
Figure 40 – Complex type "Class" 33
Figure 41 – Complex type "Function" 34
Figure 42 – Complex type "Interface" 35
Figure 43 – Complex type "Action" 35
Figure 44 – Complex type "NamedTransition" 36
Figure 45 – Complex type "MethodPrototype" 37
Figure 46 – Complex type "Method" 38
Figure 47 – Complex type "ParameterSet" 39
Figure 48 – Complex type "VarListWithAccessSpec" 40
Figure 49 – Complex type "Body" 41
Figure 50 – Complex type "BodyWithoutSFC" 41
Figure 51 – Complex type "Predicate" 42
Figure 52 – Complex type "VarList" 42
Figure 53 – Complex type "ExternalVarList" 43
Figure 54 – Complex type "VariableDecl" 43
Figure 55 – Complex type "VariableDeclPlain" 44
Figure 56 – Complex type "TypeRef" 44

Figure 57 – Complex type "Value" 45
Figure 58 – Complex type "AddressExpression" 45
Figure 59 – Complex type "FixedAddressExpression" 46
Figure 60 – Complex type "IL" 46
Figure 61 – Complex type "ST" 46
Figure 62 – Complex type "FBD" 47
Figure 63 – Complex type "FbdNetwork" 47
Figure 64 – Complex type "LD" 47
Figure 65 – Complex type "LadderRung" 48
Figure 66 – Complex type "SFC" 48
Figure 67 – Complex type "Comment" 49
Figure 68 – Complex type "Connector" 49
Figure 69 – Complex type "Continuation" 50
Figure 70 – Complex type "ActionBlocks" 51
Figure 71 – Complex type "Block" 53
Figure 72 – Complex type "DataSource" 55
Figure 73 – Complex type "DataSink" 56
Figure 74 – Complex type "Unconnected" 56
Figure 75 – Complex type "Jump" 57
Figure 76 – Complex type "Return" 57
Figure 77 – Complex type "LeftPowerRail" 58
Figure 78 – Complex type "RightPowerRail" 58
Figure 79 – Complex type "Coil" 59
Figure 80 – Complex type "Contact" 60
Figure 81 – Complex type "CompareContact" 61
Figure 82 – Complex type "Step" 62
Figure 83 – Complex type "Transition" 63
Figure 84 – Complex type "SelectionDivergence" 64
Figure 85 – Complex type "SelectionConvergence" 65
Figure 86 – Complex type "SimultaneousDivergence" 65
Figure 87 – Complex type "SimultaneousConvergence" 66
Figure 88 – Complex type "ConnectionPointIn" 67
Figure 89 – Complex type "Connection" 67
Figure 90 – Complex type "FeedbackConnection" 68
Figure 91 – Complex type "ConnectionPointOut" 68
Figure 92 – Complex type "StandardTask" 69
Figure 93 – Complex type "ParameterAssignment" 69
Figure 94 – Complex type "XyDecimalValue" 70
Figure 95 – Complex type "AddData" 70
Figure 96 – Complex type "TextBase" 70
Figure 97 – Complex type "SimpleText" 70
Figure B.1 – Only IEC 61131–3 features 93
Figure B.2 – Vendor specific extensions "AddData" 93
Figure B.3 – Vendor specific extensions (abstract complex type) 94

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/1147/FDIS	65B/1153/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.

Australian Standard®

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.

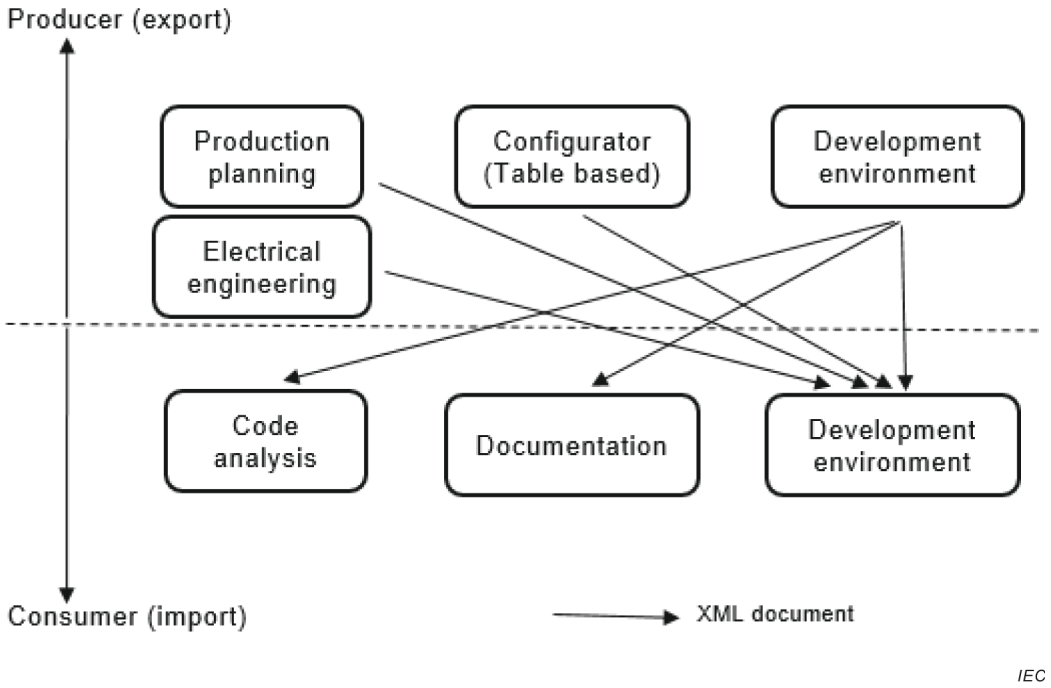

Figure 1 — Main overview of XML exchange format usage (example)

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.