

Function blocks —

Part 2: Software tools requirements

The European Standard EN 61499-2:2005 has the status of a British Standard

ICS 25.040.40; 35.240.50

National foreword

This British Standard is the official English language version of EN 61499-2:2005. It is identical with IEC 61499-2:2005.

The UK participation in its preparation was entrusted to Technical Committee GEL/65, Measurement and control, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 41 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2005

© BSI 31 October 2005

ISBN 0 580 46708 2

Amendments issued since publication

Amd. No.	Date	Comments

EUROPEAN STANDARD

EN 61499-2

NORME EUROPÉENNE

EUROPÄISCHE NORM

August 2005

ICS 25.040.40; 35.240.50

English version

Function blocks
Part 2: Software tools requirements
(IEC 61499-2:2005)

Blocs fonctionnels
Partie 2: Spécifications des outils logiciels
(CEI 61499-2:2005)

Funktionsbausteine für industrielle
Leitsysteme
Teil 2: Anforderungen an Software-
Werkzeuge
(IEC 61499-2:2005)

This European Standard was approved by CENELEC on 2005-06-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

© 2005 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Ref. No. EN 61499-2:2005 E



Documents
Document Sharing Hub

Foreword

The text of the International Standard IEC 61499-2:2005, prepared by IEC TC 65, Industrial-process measurement and control, was submitted to the formal vote and was approved by CENELEC as EN 61499-2 on 2005-06-01 without any modification.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2006-06-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2008-06-01

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 61499-2:2005 was approved by CENELEC as a European Standard without any modification.

CONTENTS

FOREWORD.....	2
INTRODUCTION	4
1 Scope	5
2 Normative references	5
3 Terms and definitions	5
4 Software tool requirements	6
4.1 Information to be provided by the software tool supplier	6
4.2 Exchange of library elements	6
4.3 Information to be provided by the supplier of library elements	6
4.4 Display of declarations.....	7
4.5 Modification of declarations	7
4.6 Validation of declarations.....	7
4.7 Implementation of declarations	7
4.8 System operation, testing and maintenance	7
Annex A (normative) Document Type Definitions (DTDs)	8
Annex B (informative) Graphics model.....	24
Annex C (informative) Examples	27
Annex ZA (normative) Normative references to international publications with their corresponding European publications	41
Figure B.1 – Graphics model.....	24
Figure B.2 – ECC drawing example	26
Table A.1 – Document Type Definition (DTD) elements.....	8
Table A.2 – DataType DTD	9
Table A.3 – DataType DTD Elements	10
Table A.4 – Library Element DTD	13
Table A.5 – LibraryElement DTD Elements	18

INTRODUCTION

The IEC 61499 series consists of four Parts:

- Part 1 contains:
 - general requirements, including an introduction, scope, normative references, definitions, and reference models;
 - rules for the declaration of *function block types*, and rules for the behaviour of *instances* of the types so declared;
 - rules for the use of function blocks in the *configuration* of distributed Industrial-Process Measurement and Control *Systems* (IPMCSs);
 - rules for the use of function blocks in meeting the communication requirements of distributed IPMCSs;
 - rules for the use of function blocks in the management of *applications*, *resources* and *devices* in distributed IPMCSs.
- Part 2 (this part of IEC 61499) defines requirements for *software tools* to support the following systems engineering tasks enumerated in Clause 1 of IEC 61499-1:
 - the specification of *function block types*;
 - the functional specification of *resource types* and *device types*;
 - the specification, analysis, and validation of distributed IPMCSs;
 - the *configuration*, *implementation*, operation, and maintenance of distributed IPMCSs;
 - the exchange of *information* among *software tools*.

It is assumed that such software tools may be used in the context of an Engineering Support System (ESS) as described in Clause C.1 of IEC 61499-1.

- Part 3 has the purpose of increasing the understanding, acceptance, and both generic and domain-specific applicability of IPMCS architectures and software tools meeting the requirements of the other Parts, by providing:
 - answers to Frequently Asked Questions (FAQs) regarding the IEC 61499 series;
 - examples of the use of IEC 61499 constructs to solve frequently encountered problems in control and automation engineering.
- Part 4 defines rules for the development of *compliance profiles* which specify the features of IEC 61499-1 and 61499-2 to be implemented in order to promote the following attributes of IEC 61499-based systems, devices and software tools:
 - interoperability of devices from multiple suppliers;
 - portability of software between software tools of multiple suppliers; and
 - configurability of devices from multiple vendors by software tools of multiple suppliers.

FUNCTION BLOCKS –

Part 2: Software tool requirements

1 Scope

This part of IEC 61499 defines requirements for *software tools* to support the following systems engineering tasks enumerated in Clause 1 of IEC 61499-1:

- the specification of *function block types*;
- the functional specification of *resource types* and *device types*;
- the specification, analysis, and validation of distributed IPMCSs;
- the *configuration, implementation*, operation, and maintenance of distributed IPMCSs;
- the exchange of *information* among *software tools*.

It is assumed that such software tools may be used in the context of an Engineering Support System (ESS) as described in Clause C.1 of IEC 61499-1.

It is beyond the scope of this part of IEC 61499 to specify the entire life cycle of industrial-process measurement and control systems (IPMCSs), or the entire set of tasks and activities required to support an IPCMS over its life cycle. However, other standards which do specify such tasks and activities may extend or modify the requirements specified in this Part.

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.

IEC 61499-1, *Function blocks - Part 1: Architecture*

IEC 61499-4, *Function Blocks - Part 4: Rules for compliance profiles*²

The normative references given in IEC 61499-1 apply to this part of IEC 61499.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61499-1 as well as the following apply.

3.1

library element

collection of declarations applying to a data type, function block type, adapter type, subapplication type, resource type, device type, or system configuration.

² To be published.

4 Software tool requirements

4.1 Information to be provided by the software tool supplier

This Clause defines the functional requirements of *software tools* that support the performance of the systems engineering tasks enumerated in Clause 1.

The supplier of a *software tool* shall specify the following information in addition to other information required in this Clause:

- a) The type or types of *library element* to which the software tool applies.
- b) The engineering task or tasks supported by the software tool. Task descriptions may be taken from the enumeration of engineering tasks given in Clause 1, or may be defined by the supplier.

4.2 Exchange of library elements

A *software tool* shall be capable of exchanging its *library elements* with other software tools. This exchange shall take the form of *data* in the format defined in Annex A, written on physical media or exchanged over communication links or networks.

4.3 Information to be provided by the supplier of library elements

NOTE The provisions of this subclause are intended to provide the means by which the provider of a library element may achieve protection of intellectual property while still providing sufficient information to permit the effective use of the library element.

The provider of a *library element* may elect to provide an *implementation* of the library element.

EXAMPLE 1 The provider of a *function block type* library element may provide an implementation of the function block type as:

- one or more *instances* of the function block type in a *resource* contained in a *device* of Class 0 or higher as described in IEC 61499-4;
- an instantiable implementation of the function block type in a *resource* contained in a *device* of Class 1 or higher as described in IEC 61499-4;
- a file in an implementation-dependent format suitable for installation in a *resource* contained in a *device* of Class 2 as described in IEC 61499-4, for instance using an XML syntax which may be defined in a compliance profile developed according to the rules given in IEC 61499-4.

When an implementation of a library element is provided, the provider is not required to provide full details of the implementation. However, the provider shall provide sufficient information to enable the user to fully determine the functionality of the provided library element.

EXAMPLE 2 The requirement of the above paragraph would be met by the provider of an *instance* of a function block type in a *resource* through the provision, at a minimum, of the following information:

- a *function block type* library element specifying its *event* and *data interfaces* as defined in IEC 61499-1, 5.2.1, and its *services* as defined in IEC 61499-1, 6.1.3;
- *resource type* and *device type* library elements showing the occurrence and connections of the function block *instances*.

4.4 Display of declarations

A software tool shall be capable of displaying the *declarations* of its associated *library elements* in a form appropriate to the engineering task. This display may utilize the graphical or textual formats defined in IEC 61499-1, or a format defined by the supplier of the software tool.

NOTE The *declarations* of a library element may define its *interfaces* (event and data inputs and outputs) and internal *variables* as well as its *algorithms* and the control of their *execution*, for example via an *execution control chart* (ECC), etc.

4.5 Modification of declarations

A software tool shall enable its user to modify the declarations of its associated library elements as appropriate to the engineering task. Such modifications may include adding, deleting or changing the contents of declarations, and may be performed either graphically or textually or both.

4.6 Validation of declarations

If required by the associated engineering task, a software tool shall provide facilities for validation of the declarations of its associated library elements. Such facilities may include, but are not limited to:

- a) Checking the correctness of the syntax of declarations.
- b) Checking the semantic correctness of declarations, for instance, checking whether all *function block instances* in an *application* and its associated *subapplications* are properly allocated to *resources*, interconnected within resources, and intercommunicating among resources in a *system configuration*.
- c) Simulation and testing of the operation of an *instance* of a library element *type*, either by itself or in association with other instances of the same or different types.

4.7 Implementation of declarations

If required by the associated engineering task, a software tool shall provide facilities for the *implementation* of the *declarations* of its associated *library elements*. Such facilities may include, but are not limited to:

- a) The production of executable code ("firmware") for embedding in *instances* of *resource types* and *device types*.
- b) The creation and interconnection ("downloading") of *function block instances* in *resources* and *devices*, for instance by using the management facilities defined in subclause 3.3 and Annexes F and G of IEC 61499-1.

4.8 System operation, testing and maintenance

If required by the associated engineering task, a software tool shall provide facilities for the operation, testing and maintenance of an Industrial Process Measurement and Control System (IPMCS) specified by its associated library elements. Such facilities may include, but are not limited to:

- a) The facilities described in preceding subclauses of this Clause.
- b) The information exchange facilities defined in subclause 6.2 and Annex F of IEC 61499-1.

Annex A (normative)

Document Type Definitions (DTDs)

A.1 General principles

This Annex presents Document Type Definitions (DTDs) for the exchange of IEC 61499 library elements between *software tools*. These DTDs are defined in the syntax defined in the eXtensible Markup Language (XML) specification (see www.w3.org/TR/2000/REC-xml-20001006).

The correspondences between the DTD elements given in this Annex, the library elements defined in IEC 61499-1, C.2.2, and the textual syntax given in IEC 61499-1, Annex B are given in Table A.1.

Table A.1 – Document Type Definition (DTD) elements

DTD element	LibraryElement	Textual syntax
DataType	DataTypeDeclaration	data_type_declaration (IEC 61131-3-B.1.3)
FBType	FBTypeDeclaration	fb_type_declaration
SubapplicationType	SubapplicationTypeDeclaration	subapplication_type_ declaration
AdapterType	AdapterTypeDeclaration	adapter_type_declaration
ResourceType	ResourceTypeDeclaration	resource_type_ specification
DeviceType	DeviceTypeDeclaration	device_type_specification
System	SystemConfiguration	system_configuration

The first table of each subclause of this Annex contains the DTD for the corresponding library element. The second table of each subclause provides a reference to the textual syntax (if any) plus an explanation for the major elements and attributes in the DTD. Following this, examples are given of the resulting XML files for typical library elements.

If there is a conflict between the provisions of this Annex and the provisions of Annex B of IEC 61499-1, the provisions of the latter shall prevail.

NOTE 1 The examples given in this Annex provide a representative but not exhaustive sample of the features of the associated DTDs. In particular, these examples are not intended to be used as a test suite for compliance to the provisions of this part of IEC 61499.

A.2 DataType DTD

An XML document complying with the DTD in Table A.2 represents a DataTypeDeclaration object as described in Clause C.1 of IEC 61499-1.

Table A.2 – DataType DTD

<pre><?xml version="1.0" encoding="UTF-8"?> <!ELEMENT DataType (Identification?, VersionInfo+, CompilerInfo?, ASN1Tag?, (DirectlyDerivedType EnumeratedType SubrangeType ArrayType StructuredType))> <!ATTLIST DataType Name CDATA #REQUIRED Comment CDATA #IMPLIED></pre>
<pre><!ELEMENT Identification EMPTY> <!ATTLIST Identification Standard CDATA #IMPLIED Classification CDATA #IMPLIED ApplicationDomain CDATA #IMPLIED Function CDATA #IMPLIED Type CDATA #IMPLIED Description CDATA #IMPLIED></pre>
<pre><!ELEMENT VersionInfo EMPTY> <!ATTLIST VersionInfo Organization CDATA #REQUIRED Version CDATA #REQUIRED Author CDATA #REQUIRED Date CDATA #REQUIRED Remarks CDATA #IMPLIED></pre>
<pre><!ELEMENT ASN1Tag EMPTY> <!ATTLIST ASN1Tag Class (UNIVERSAL APPLICATION CONTEXT PRIVATE) #IMPLIED Number CDATA #REQUIRED></pre>
<pre><!ELEMENT CompilerInfo (Compiler*)> <!ATTLIST CompilerInfo header CDATA #IMPLIED classdef CDATA #IMPLIED></pre>
<pre><!ELEMENT Compiler EMPTY> <!ATTLIST Compiler Language (Java Cpp C Other) #REQUIRED Vendor CDATA #REQUIRED Product CDATA #REQUIRED Version CDATA #REQUIRED></pre>
<pre><!ELEMENT DirectlyDerivedType EMPTY> <!ATTLIST DirectlyDerivedType BaseType (BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TIME_OF_DAY TOD DATE_AND_TIME DT STRING BYTE WORD DWORD LWORD WSTRING) #REQUIRED InitialValue CDATA #IMPLIED Comment CDATA #IMPLIED></pre>
<pre><!ELEMENT EnumeratedType (EnumeratedValue+)> <!ATTLIST EnumeratedType InitialValue CDATA #IMPLIED Comment CDATA #IMPLIED></pre>
<pre><!ELEMENT EnumeratedValue EMPTY> <!ATTLIST EnumeratedValue Name CDATA #REQUIRED Comment CDATA #IMPLIED></pre>
<pre><!ELEMENT SubrangeType (Subrange)> <!ATTLIST SubrangeType BaseType (SINT INT DINT LINT USINT UINT UDINT ULINT) #REQUIRED InitialValue CDATA #IMPLIED Comment CDATA #IMPLIED></pre>

Table A.2 – DataType DTD

<pre><!ELEMENT Subrange EMPTY> <!--ATTLIST Subrange LowerLimit CDATA #REQUIRED UpperLimit CDATA #REQUIRED--></pre>
<pre><!ELEMENT ArrayType (Subrange+)> <!--ATTLIST ArrayType BaseType CDATA #REQUIRED InitialValues CDATA #IMPLIED Comment CDATA #IMPLIED--></pre>
<pre><!--ELEMENT StructuredType (VarDeclaration SubrangeVarDeclaration)+> <!--ATTLIST StructuredType Comment CDATA #IMPLIED--></pre>
<pre><!--ELEMENT VarDeclaration EMPTY > <!--ATTLIST VarDeclaration Name CDATA #REQUIRED Type CDATA #REQUIRED ArraySize CDATA #IMPLIED InitialValue CDATA #IMPLIED Comment CDATA #IMPLIED--></pre>
<pre><!--ELEMENT SubrangeVarDeclaration (Subrange+) > <!--ATTLIST SubrangeVarDeclaration Name CDATA #REQUIRED Type (SINT INT DINT LINT USINT UINT UDINT ULINT) #REQUIRED InitialValue CDATA #IMPLIED Comment CDATA #IMPLIED--></pre>

Explanations of the elements of the DTD given in Table A.2, and (where applicable) references to the formal syntax for their attributes, are given in Table A.3.

Table A.3 – DataType DTD elements

Element attributes	Textual syntax (IEC 61131-3, Annex B)		Explanation
DataType			See IEC 61131-3
Name	data_type_name		
Comment	--	A comment as per IEC 61131-3 without (* and *) delimiters	
Identification	Information for data base retrieval		
Standard	Primary reference standard in number-part-subclause format		
Classification	Classification code as defined in reference standard		
ApplicationDomain	Application domain as defined in reference standard		
Function	Function of this element as defined in reference standard		
Type	Element type (e.g., device type) as defined in reference standard		
Description	Descriptive phrase as defined in reference standard		

Table A.3 – DataType DTD elements

Element attributes	Textual syntax (IEC 61131-3, Annex B)		Explanation
VersionInfo	--	Possibly one of several entries: First entry – Most recent version Second entry – Immediately preceding released version, etc. last entry – First released version	
Organization	--	The organization supplying this library element	
Version	digit [digit] '.' digit [digit] [letter]		The Version identification for this library element
Author	--	The author of this library element	
Date	date_literal ['-' daytime]		The release date of this version
Remarks	--	Comments relating to this version	
ASN1Tag	ASN.1 tag per ISO/IEC 8824-1		
Class	ASN.1 tag class per ISO/IEC 8824-1		
Number	ASN.1 tag number per ISO/IEC 8824-1		
CompilerInfo	--	Information for and about compilers used with this class	
header	--	Header information such as package, imports, etc.	
classdef	--	The class definition information such as superclass and implemented interfaces. If none is given, a default abstract superclass is used.	
Compiler	--	Possibly one of several compilers used with this version	
Language	--	The source language of this compiler	
Vendor	--	The vendor of this compiler	
Product	--	The product name of this compiler	
Version	--	The version of this compiler	
DirectlyDerivedType	See IEC 61131-3, Tables 12 and 14, #1		
BaseType	elementary_type_name		
InitialValue	constant		
Comment	A comment as per IEC 61131-3 without (* and *) delimiters		
EnumeratedType	See IEC 61131-3, Tables 12 and 14, #2		
InitialValue	enumerated_value		
Comment	A comment as per IEC 61131-3 without (* and *) delimiters		
EnumeratedValue	See IEC 61131-3, Table 14, #2		
Name	enumerated_value		
Comment	A comment as per IEC 61131-3 without (* and *) delimiters		
SubrangeType	--	See IEC 61131-3, Tables 12 and 14, #3	
BaseType	integer_type_name		
InitialValue	signed_integer		
Comment	A comment as per IEC 61131-3 without (* and *) delimiters		

Table A.3 – DataType DTD elements

Element attributes	Textual syntax (IEC 61131-3, Annex B)	Explanation
Subrange	See IEC 61131-3, Tables 12 and 14, #3	
LowerLimit	signed_integer	
UpperLimit	signed_integer	
ArrayType	See IEC 61131-3, Tables 12 and 14, #4	
BaseType	non_generic_type_name	
InitialValues	array_initialization	
StructuredType	See IEC 61131-3, Tables 12, #5 and 14, #5 and #6	
VarDeclaration		
Name	structure_element_name	
Type	non_generic_type_name	
ArraySize	See NOTE 1	
InitialValue	See NOTE 2	
Comment	A comment as per IEC 61131-3 without (* and *) delimiters	
SubrangeVarDeclaration	See IEC 61131-3, 2.3.3.	
Name	structure_element_name	
Type	integer_type_name	
InitialValue	signed_integer	
Comment	A comment as per IEC 61131-3 without (* and *) delimiters	
NOTE 1 The syntax of this attribute when present is equivalent to the syntactic expression (subrange {' , ' subrange}) integer {' , ' integer} where the non-terminals subrange and integer are as defined in Annex B of IEC 61131-3. Each term of the second form is equivalent to the subrange 0 . . n-1, where n is the value of the corresponding integer syntactic element. If this attribute is missing, the structure component is not an anonymously defined array.		
NOTE 2 The syntax of this attribute is the syntax for initialization of the corresponding variable type as defined in Annex B.1.4.3 of IEC 61131-3.		

EXAMPLE The structured data type ANALOG_CHANNEL_CONFIGURATIONI example is expressed in IEC 61131-3, Table 14 as follows:

```

TYPE ANALOG_CHANNEL_CONFIGURATIONI :
STRUCT
  RANGE : ANALOG_SIGNAL_RANGE ;
  MIN_SCALE : ANALOG_DATA := -4095 ;
  MAX_SCALE : ANALOG_DATA := 4095 ;
END_STRUCT ;
END_TYPE

```