

Australian Standard™

**Industrial automation systems and
integration—Parts library**

**Part 42: Description methodology:
Methodology for structuring part
families**



This Australian Standard was prepared by Committee IT-006, Information Technology for Industrial Automation and Integration. It was approved on behalf of the Council of Standards Australia on 23 March 2004 and published on 3 June 2004.

The following are represented on Committee IT-006:

Association of Consulting Engineers Australia
Australian Electrical and Electronic Manufacturers Association
CSIRO Centre for Planning and Design
CSIRO Manufacturing & Infrastructure Technology
Department of Defence (Australia)
Institute of Instrumentation, Control and Automation Australia
Institution of Engineers Australia
Monash University
RMIT University
The University of Melbourne

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at www.standards.com.au and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year, and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at mail@standards.org.au, or write to the Chief Executive, Standards Australia International Ltd, GPO Box 5420, Sydney, NSW 2001.

This Standard was issued in draft form for comment as DR 04075.

Australian Standard™

Industrial automation systems and integration—Parts library

Part 42: Description methodology: Methodology for structuring part families

First published as AS ISO 13584.42—2004.

COPYRIGHT

© Standards Australia International

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia International Ltd
GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 5911 4

 **Documents**
Document Sharing Hub

PREFACE

This Standard was prepared by the Standards Australia Committee IT-006, Information Technology for Industrial Automation and Integration.

This Standard is identical with, and has been reproduced from, ISO 13584-42:1998, *Industrial automation systems and integration—Parts library, Part 42: Description methodology: Methodology for structuring part families*.

The objective of this Standard is to provide the principles that shall be used for defining families of parts and properties of parts to fully characterize parts and associated properties.

This Standard is Part 42 of AS ISO 13584, *Industrial automation systems and integration—Parts library*, which is published in parts as follows:

Part 1: Overview and fundamental principles

Part 101: Geometrical view exchange protocol by parametric program

Part 20: Logical resource: Logical model of expressions

Part 24: Logical resource: Logical model of supplier library

Part 26: Logical resource: Information supplier identification

Part 31: Implementation resources: Geometric programming interface

Part 42: Description methodology: Methodology for structuring part families (this Standard)

The terms ‘normative’ and ‘informative’ are used to define the application of the annex to which they apply. A normative annex is an integral part of a standard, whereas an informative annex is only for information and guidance.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover
- (b) In the source text ‘this International Standard’ should read ‘this Australian Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>		<i>Australian Standard</i>	
ISO/IEC		AS/NZS	
8824-1	Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation	8824.1	Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation
ISO		AS/NZS	
8601	Data elements and interchange formats—Information interchange—Representations of dates and times	3802	Data elements and interchange formats—Information interchange—Representations of dates and times
9735	Electronic data interchange for administration, commerce and transport (EDIFACT)—Application level syntax rules	3801	Electronic data interchange for administration, commerce and transport (EDIFACT)—Application level syntax rules
ISO		AS	
31-0	Quantities and units—Part 0: General principles	2900.0	Quantities and units, Part 0: General principles
4217	Codes for the representation of names of currencies and funds	3759	Codes for the representation of names of currencies and funds

12083	Information and documentation— Electronic manuscript preparation and markup	4449	Information and documentation— Electronic manuscript preparation and markup
6093	Information processing— Representation of numerical values in character strings for information interchange	3767	Information processing— Representation of numerical values in character strings for information interchange
8879	Information processing—Text and office systems—Standards Generalized Markup Language (SGML)	3514	Information processing—Text and office systems—Standards Generalized Markup Language (SGML)
10303-1	Industrial automation systems and integration—Product data representation and exchange—Part 1: Overview and fundamental principles	10303.1	Industrial automation systems and integration—Product data representation and exchange, Part 1: Overview and fundamental principles
10303-11	Industrial automation systems and integration—Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual	10303.11	Industrial automation systems and integration—Product data representation and exchange, Part 11: Description methods: The EXPRESS language reference manual
10303-21	Industrial automation systems and integration—Product data representation and exchange—Part 21: Implementation methods: Clear text encoding of the exchange structure	10303.21	Industrial automation systems and integration—Product data representation and exchange, Part 21: Implementation methods: Clear text encoding of the exchange structure
10303-41	Industrial automation systems and integration—Product data representation and exchange—Part 41: Fundamentals of product description and support	10303.41	Industrial automation systems and integration—Product data representation and exchange, Part 41: Fundamentals of product description and support
10303-42	Industrial automation systems and integration—Product data representation and exchange—Part 42: Integrated generic resources: Geometric and topological representation	10303.42	Industrial automation systems and integration—Product data representation and exchange, Part 42: Integrated generic resources: Geometric and topological representation

CONTENTS

	<i>Page</i>
Introduction	x
1 Scope	1
2 Normative references.....	2
3 Terms and definitions.....	3
3.1 Terms and definitions from ISO 10303-1	3
3.2 Terms and definitions from ISO 13584-1	4
3.3 Terms and definitions from ISO 13584-10	4
3.4 Other terms and definitions.....	4
4 Abbreviated terms	6
5 Description of a hierarchy of families of parts	6
5.1 Simultaneous description of families of parts and parts properties.....	6
5.2 Applicable and visible properties.....	7
5.3 Purpose of the standardised identification hierarchy	7
5.4 Use of the standardised identification hierarchy	7
5.5 Class valued property	7
5.6 Extensions of the common dictionary schema addressing IEC 61360 needs	8
5.7 Extension of the common dictionary schema addressing ISO 13584 needs.....	8
5.8 Compatibility between IEC 61360 and ISO 13584.....	8
6 Rules for creating hierarchies of families of parts	9
6.1 Choice of family hierarchy.....	9
6.1.1 Field of application	9
6.1.2 Upper section of the hierarchy	9
6.1.3 Lower section of the hierarchy	10
6.1.4 Simple family of parts.....	10
6.1.5 Multiple perspectives on the hierarchy.....	10
6.2 Association of properties.....	11
6.2.1 Properties to be considered.....	11
6.2.2 Semantic identification of properties	11
6.2.3 Factoring rule	11
7 Dictionary elements that describe properties of parts	12
7.1 Mapping of properties onto the common dictionary schema	12
7.2 Attributes	12
7.2.1 Code	13
7.2.2 Definition Class	14
7.2.3 Data Type	14
7.2.4 Preferred Name	14
7.2.5 Short Name	15

	<i>Page</i>
7.2.6 Preferred Letter Symbol.....	15
7.2.7 Synonymous Letter Symbol.....	15
7.2.8 Synonymous Name.....	15
7.2.9 Property Type Classification	16
7.2.10 Definition	16
7.2.11 Source Document of Definition	16
7.2.12 Note	17
7.2.13 Remark	17
7.2.14 Unit.....	17
7.2.15 Condition	18
7.2.16 Formula.....	18
7.2.17 Value Format	18
7.2.18 Date of Original Definition	20
7.2.19 Date of Current Version	20
7.2.20 Date of Current Revision.....	20
7.2.21 Version Number	20
7.2.22 Revision Number	21
7.3 Rules for defining new versions and/or revision of properties	21
7.3.1 Changes in the attribute of properties.....	21
7.3.2 Propagated changes in property version numbers	22
8 Dictionary elements that describe families of parts.....	23
8.1 Mapping of families onto the common dictionary schema	23
8.2 Attributes.....	23
8.2.1 Code	24
8.2.2 Superclass	24
8.2.3 Preferred Name	25
8.2.4 Short Name.....	25
8.2.5 Synonymous name	25
8.2.6 Visible Types.....	26
8.2.7 Applicable Types.....	26
8.2.8 Sub-class Selection Properties	27
8.2.9 Visible Properties	27
8.2.10 Applicable Properties	27
8.2.11 Class Value Assignment.....	28
8.2.12 Definition	28
8.2.13 Source Document of Definition	28
8.2.14 Note	28
8.2.15 Remark	29
8.2.16 Simplified Drawing	29
8.2.17 Date of Original Definition	29
8.2.18 Date of Current Version	30
8.2.19 Date of Current Revision.....	30
8.2.20 Version Number	30
8.2.21 Revision Number	30
8.3 Rules for defining new version or revision of classes	31
8.3.1 Change in the attribute of class	31
8.3.2 Propagated changes in family version numbers	32
9 Bibliography.....	34
Annex A (normative) Survey of type classification codes of non-quantitative data element types	35
Annex B (normative) Short names of entities.....	36
Annex C (normative) Information object registration.....	38
C.1 Document identification	38

	<i>Page</i>
C.2 Schema identification.....	38
C.2.1 ISO13584_IEC61360_dictionary_schema	38
C.2.2 ISO13584_IEC61360_language_resource_schema.....	38
Annex D (informative) Common IEC/ISO dictionary schema.....	39
D.1 General.....	39
D.1.1 Scope and object of the common dictionary schema	39
D.1.2 Interoperability of ISO 13584 and IEC 61360.....	40
D.2 Overview of the dictionary schema.....	40
D.3 ISO13584_IEC61360_dictionary_schema	41
D.3.1 References to other schemata	42
D.3.2 Constant definitions	42
D.3.3 Basic Semantic Units: defining and using the dictionary	43
D.3.3.1 Requirements for exchange	43
D.3.3.2 Three levels architecture of the dictionary data	43
D.3.3.2.1 Basic_semantic_unit	44
D.3.3.2.2 Dictionary_element	45
D.3.3.2.3 Content_item.....	46
D.3.3.3 Overview of basic semantic units and dictionary elements.....	47
D.3.3.4 Identification of dictionary elements: three levels structure.....	47
D.3.3.5 Extension possibilities for other types of data	48
D.3.3.5.1 Supplier_related_BSU.....	48
D.3.3.5.2 Class_related_BSU.....	48
D.3.3.5.3 Supplier_BSU_relationship	48
D.3.3.5.4 Class_BSU_relationship	49
D.3.4 Supplier Data.....	49
D.3.4.1 Supplier_BSU.....	49
D.3.4.2 Supplier_element	50
D.3.5 Class Data.....	51
D.3.5.1 General.....	51
D.3.5.1.1 Class_BSU.....	51
D.3.5.1.2 Class_and_property_elements	53
D.3.5.1.3 Class	54
D.3.5.2 Item_class	56
D.3.5.3 Component_class.....	57
D.3.5.4 Material_class	57
D.3.6 Data Element Type / properties data.....	58
D.3.6.1 Property_BSU	58
D.3.6.2 Property_DET.....	59
D.3.6.3 Condition, dependent and non-dependent Data Element Types	60
D.3.6.3.1 Condition_DET	61
D.3.6.3.2 Dependent_P_DET	61
D.3.6.3.3 Non_dependent_P_DET	61
D.3.6.4 Class-valued properties	62
D.3.7 Domain data: the type system.....	63
D.3.7.1 General.....	63
D.3.7.1.1 Data_type_BSU	63
D.3.7.1.2 Data_type_element.....	64
D.3.7.2 The type system.....	65
D.3.7.2.1 Data_type.....	65
D.3.7.2.2 Simple_type	65
D.3.7.2.3 Number_type.....	66
D.3.7.2.4 Int_type	66
D.3.7.2.5 Int_measure_type	66

	<i>Page</i>
D.3.7.2.6 Int_currency_type.....	67
D.3.7.2.7 Non_quantitative_int_type.....	67
D.3.7.2.8 Real_type.....	67
D.3.7.2.9 Real_measure_type.....	68
D.3.7.2.10 Real_currency_type.....	68
D.3.7.2.11 Boolean_type.....	68
D.3.7.2.12 String_type.....	69
D.3.7.2.13 Non_quantitative_code_type.....	69
D.3.7.2.14 Complex_type.....	69
D.3.7.2.15 Level_type.....	70
D.3.7.2.16 Level.....	70
D.3.7.2.17 Class_instance_type.....	71
D.3.7.2.18 Entity_instance_type.....	71
D.3.7.2.19 Placement_type.....	71
D.3.7.2.20 Axis1_placement_type.....	72
D.3.7.2.21 Axis2_placement_2d_type.....	72
D.3.7.2.22 Axis2_placement_3d_type.....	73
D.3.7.2.23 Named_type.....	73
D.3.7.3 Values.....	73
D.3.7.3.1 Value_domain.....	74
D.3.7.3.2 Value_type.....	75
D.3.7.3.3 Dic_value.....	75
D.3.7.4 Extension to ISO 10303-41 unit definitions.....	75
D.3.7.4.1 Non_si_unit.....	75
D.3.7.4.2 Assert_ONEOF rule.....	76
D.3.7.4.3 Dic_unit.....	76
D.3.8 Basic type and entity definitions.....	77
D.3.8.1 Basic type definitions.....	77
D.3.8.1.1 Class_code_type.....	77
D.3.8.1.2 Code_type.....	77
D.3.8.1.3 Currency_code.....	78
D.3.8.1.4 Date_type.....	78
D.3.8.1.5 Definition_type.....	79
D.3.8.1.6 DET_classification_type.....	79
D.3.8.1.7 Data_type_code_type.....	79
D.3.8.1.8 Note_type.....	80
D.3.8.1.9 Pref_name_type.....	80
D.3.8.1.10 Property_code_type.....	80
D.3.8.1.11 Remark_type.....	80
D.3.8.1.12 Revision_type.....	81
D.3.8.1.13 Short_name_type.....	81
D.3.8.1.14 Supplier_code_type.....	81
D.3.8.1.15 Syn_name_type.....	82
D.3.8.1.16 Value_code_type.....	82
D.3.8.1.17 Value_format_type.....	82
D.3.8.1.18 Version_type.....	83
D.3.8.1.19 Source_doc_type.....	83
D.3.8.2 Basic entity definitions.....	83
D.3.8.2.1 Dates.....	83
D.3.8.2.2 Document.....	84
D.3.8.2.3 Graphics.....	84
D.3.8.2.4 Identified_document.....	84
D.3.8.2.5 Item_names.....	85
D.3.8.2.6 Label_with_language.....	86
D.3.8.2.7 Mathematical_string.....	86
D.3.9 Function definitions.....	87
D.3.9.1 Acyclic_superclass_relationship function.....	87

	<i>Page</i>
D.3.9.2 At_most_two_synonyms_per_language function.....	87
D.3.9.3 Check_syn_length function	88
D.3.9.4 Codes_are_unique function	88
D.3.9.5 Definition_available_implies function.....	89
D.3.9.6 Is_subclass function.....	89
D.3.9.7 String_for_derived_unit function.....	90
D.3.9.8 String_for_named_unit function	91
D.3.9.9 String_for_SI_unit function	92
D.3.9.10 String_for_unit function	93
D.3.9.11 All_class_descriptions_reachable function.....	94
D.3.9.12 Compute_known_visible_properties function.....	94
D.3.9.13 Compute_known_visible_data_types function	95
D.3.9.14 Compute_known_applicable_properties function.....	96
D.3.9.15 Compute_known_applicable_data_types function	97
D.3.9.16 List_to_set.....	97
D.4 ISO13584_IEC61360_language_resource_schema.....	98
D.4.1 ISO13584_IEC61360_language_resource_schema type and entity definitions	99
D.4.1.1 Language_code.....	99
D.4.1.2 Global_language_assignment.....	99
D.4.1.3 Present_translations.....	99
D.4.1.4 Translatable_label	100
D.4.1.5 Translated_label.....	100
D.4.1.6 Translatable_text.....	100
D.4.1.7 Translated_text.....	101
D.4.2 ISO13584_IEC61360_language_resource_schema function definitions	101
D.4.2.1 Check_label_length function	101
D.4.2.2 Check_text_length function.....	102
D.4.3 ISO13584_IEC61360_language_resource_schema rule definition	102
D.5 Example Physical File	103
D.5.1 Some example data.....	103
D.5.1.1 Supplier data	103
D.5.1.2 Root class data.....	103
D.5.1.3 Material data.....	104
D.5.1.4 Component data.....	104
D.5.1.5 Electric / Electronic component data	105
D.5.2 Templates derived from the EXPRESS code	106
D.5.2.1 Templates for supplier data.....	107
D.5.2.2 Templates for class data	107
D.5.2.3 Templates for property_DET data.....	108
D.5.2.4 Templates for data type data.....	108
Annex E (informative) Survey of main classes and categories of properties.....	110
Annex F (informative) Survey of type classification codes of quantitative data element types.....	111
Annex G (informative) EXPRESS-G diagrams	119
Annex H (informative) Definitions from ISO 13584-1.....	127
Annex J (informative) Definitions from ISO 13584-10.....	128
Index	129

Figures

Figure D.1 — Overview of the dictionary schema.....	41
Figure D.2 — Pieces of data with relationships	43
Figure D.3 — Implementation of "inter-piece" relationships using basic semantic units.....	44
Figure D.4 — Relationship between basic semantic unit and dictionary element.....	46
Figure D.5 — Current BSUs and dictionary elements.....	47
Figure D.6 — Overview of supplier data and relationships	49
Figure D.7 — Overview of class data and relationships	51
Figure D.8 — Overview of property data element type data and relationships.....	60
Figure D.9 — Kinds of data element types	60
Figure D.10 — Entity hierarchy for the type system	63
Figure D.11 — Overview of non-quantitative data element Types.....	74
Figure D.12 — ISO13584_IEC61360_language_resource_schema and support_resource_schema	98

Tables

Table 1 — Overview of versioning in property updating operations.....	22
Table 2 — Propagated changes in version numbers of properties.....	23
Table 3 — Overview of versioning in class updating operations.....	32
Table B.1 — Short names of entities	36

INTRODUCTION

ISO 13584 is an International Standard for the computer-interpretable representation and exchange of part library data. The objective is to provide a neutral mechanism capable of transferring parts library data, independent of any application that is using a parts library data system. The nature of this description makes it suitable not only for the exchange of files containing parts, but also as a basis for implementing and sharing databases of parts library data.

ISO 13854 is organised as a series of parts, each published separately. The parts of ISO 13854 fall into one of the following series: conceptual descriptions, logical resources, implementation resources, description methodology, conformance testing, view exchange protocol, and standardised content. The series are described in ISO 13584-1. This part of ISO 13584 is a member of the description methodology series.

This part of ISO 13584 provides rules and guidelines for library data suppliers to create hierarchies of families of parts according to a common methodology intended to enable multi-supplier consistency. These rules pertain to the following: the method for grouping parts into families of parts to form a hierarchy; the dictionary elements that describe the families and properties of parts.

This part of ISO 13584 refers as a normative reference to the data model that specifies the exchange of dictionary data. This EXPRESS specification was developed as a common model for ISO 13584 and IEC 61360. It is intended to be published as IEC 61360-2. For convenience this common model is provided in this part of ISO 13584 as an informative annex that duplicates the normative content of IEC 61360-2. This part of ISO 13584 also provides the mapping of the concepts described here onto the common model.

AUSTRALIAN STANDARD

Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families

1 Scope

This part of ISO 13584 specifies the principles that shall be used for defining families of parts and properties of parts to fully characterize parts and associated properties.

The rules and guidelines provided in this part of ISO 13584 are mandatory for the standardisation committees responsible for creating standardised identification hierarchies.

The use of these rules by suppliers and users is recommended as a methodology for building their own hierarchies.

The following are within the scope of this part of ISO 13584:

- the rule to group parts into generic families of parts and simple families of parts;
- the rules for the choice of the appropriate properties that shall be associated with the families of parts;
- the attributes that shall be provided by library data suppliers to describe the families and properties of parts.
- the specifications of those attributes in the EXPRESS information model that provide for the exchange of such dictionary data.

NOTE 1 The EXPRESS information model for the exchange of dictionary data is defined in IEC 61360-2.

NOTE 2 The content of this EXPRESS information model is in the informative Annex D of this part of ISO 13584 that duplicates the normative content of IEC 61360-3.

The following are outside the scope of this part of ISO 13584:

- libraries of assembled parts with aggregate structure (Level 3 libraries);
- the description of the parts themselves;
- the descriptions of the functional models that may refer to some family of parts;
- the description of tables, program libraries and documents that may refer to some family of parts;
- the description of the systems intended to manage parts libraries; and
- the EXPRESS resource constructs for references between libraries.

NOTE The EXPRESS resource constructs for tables, programs libraries, documents and for references between libraries are defined in ISO 13584-24.

The structure of the information and the methodology defined in this part of ISO 13584 enable the following:

- integration in the same data repository of different parts libraries originating from different library data suppliers with an uniform access mechanism provided by a dictionary;
- integration in the same data repository of different definitions of materials originating from different suppliers with an uniform access mechanism provided by a dictionary;
- referencing another supplier library assumed to be available on the receiving system;
- referencing a standardised identification hierarchy when such a hierarchy exists;
- definition by an end-user of a local classification or search hierarchy, and the mapping of these hierarchies onto the supplier libraries available on its system.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 13584. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 13584 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 31-0:1992, *Quantities and units — Part 0: General principles*.

ISO 4217, *Codes for the representation of names of currencies and funds*.

ISO 6093:1985, *Information processing — Representation of numerical values in character strings for information interchange*.

ISO 8601:1988, *Data elements and interchange formats — Information interchange — Representations of dates and times*.

ISO/IEC 8824-1:1995, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

ISO 8879-1:1987, *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*.

ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*.

ISO 9735:1988, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules*.

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*.

ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-21:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303-42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 12083:1992, *Information and documentation — Electronic manuscript preparation and markup.*

ISO/IEC 11179-3:1994, *Information technology — Specification and standardization of data elements — Basic attributes of data elements.*

ISO 13584-10:—¹, *Industrial automation systems and integration — Parts library — Part 10: Conceptual model.*

International Classification of Standards (ICS), 1993.

IEC 61360-1:1995, *Standard data element types with associated classification scheme for electric components — Part 1: Definitions — Principles and methods.*

IEC 61360-2:—¹, *Standard data element types with associated classification scheme for electric components — Part 2: EXPRESS Dictionary Schema.*

IEC 61360-3:1995, *Standard data element types with associated classification scheme for electric components — Part 3: Maintenance and validation procedures.*

IEC 61360-4:1997, *Standard data element types with associated classification scheme for electric components — Part 4: IEC reference collection of standard data element types, component classes and terms.*

3 Terms and definitions

For the purposes of this part of ISO 13584, the following terms and definitions apply.

3.1 Terms and definitions from ISO 10303-1

For the purposes of this part of ISO 13584, the following terms and definitions given in ISO 10303-1:1994 (which are repeated below for convenience) apply.

3.1.1

information

facts, concepts or instructions.

¹ To be published

3.1.2

information model

a formal model of a bounded set of facts, concepts or instructions to meet a specified requirement.

3.2 Terms and definitions from ISO 13584-1

For the purposes of this part of ISO 13584, the definitions given in ISO 13584-1 for the following terms apply.

- Library data supplier;
- Part;
- Parts library;
- Resource construct;
- Supplier library.

NOTE These definitions are duplicated in the informative Annex H of this part of ISO 13584.

3.3 Terms and definitions from ISO 13584-10

For the purposes of this part of ISO 13584, the definitions given in ISO 13584-10 for the following terms apply.

- Applicable property;
- Characteristic of a part (part characteristic);
- Context dependent characteristic;
- Context parameter;
- Family of parts;
- Generic family of parts;
- Property;
- Simple family of parts;
- Visible property.

NOTE These definitions are duplicated in the informative Annex I of this part of ISO 13584.

3.4 Other terms and definitions

For the purposes of this part of ISO 13584, the following apply.

3.4.1

basic semantic unit

the entity that provides an absolute and universally unique identification of certain objects of the application domain (e.g. classes, data element types).

3.4.2

class valued property

a property that has one single value for a whole family of parts. Its value is not defined individually for every single part of that family, but for the family itself.

NOTE Class valued properties may be used to capture some commonality between different families when such a commonality is not captured by the hierarchy structure (see example of RULE 5b).

3.4.3

common dictionary schema

the information model for a dictionary, using the information modelling language EXPRESS, resulting from a joint effort between ISO TC 184/SC4/WG2 and IEC SC3D;

NOTE The common dictionary schema is specified in IEC 61360-2, and its content is provided in the informative annex D of this part of ISO 13584.

3.4.4

data element type

unit of data for which the identification, the description and value representation have been specified.

3.4.5

data type

the specification of the type of a property that specifies the allowed set of values of that property.

3.4.6

dictionary data

the set of data that describes hierarchies of families of parts and properties of these parts.

NOTE The dictionary data should be exchanged using the common dictionary schema.

3.4.7

dictionary element

the set of attributes that constitutes the dictionary description of certain objects of the application domain (e.g. classes, data element types).

3.4.8

quantitative data element type

data element type with a numerical value representing a physical quantity, a quantity of information or a count of objects.

3.4.9

non-quantitative data element type

data element type that identifies or describes an object by means of codes, abbreviations, names, references or descriptions.

3.4.10

property

an information that may be represented by a data element type.

3.4.11

standardized identification hierarchy

dictionary data that is defined by a standardisation organisation.

4 Abbreviated terms

For the purpose of this part of ISO 13584 the following abbreviated terms apply.

- BSU : Basic Semantic Unit;
- DET: Data Element Type;
- ICS: International Classification of Standards;
- SI: Système International d'Unités (International System of Units).

5 Description of a hierarchy of families of parts

5.1 Simultaneous description of families of parts and parts properties

The hierarchy shall be organised as a tree-structure with single inheritance. At each level of the hierarchy, the sub-families of a generic family of parts shall be mutually exclusive. The hierarchy shall include two sections:

- the upper section classifies technical fields and so defines the first levels of generic families of parts. These generic families of parts are used for classification purposes. They are not intended to be instantiated.
- the lower section splits these families into generic sub-families to obtain simple families of parts. In choosing these families the instantiation rule (RULE 3) is important. These families are intended to be instantiated.

Both the standardised identification hierarchy and the supplier hierarchies shall simultaneously define the following:

- (hierarchical) families of parts;
- (non hierarchical) properties.

The meaning of each family of parts and of each property shall be such that a human observer shall be able to determine, for a given part:

- those families of parts to which it belongs and those to which it does not belong;
- that characteristic of the part to which every applicable property corresponds.

The simultaneous definition of families of parts and of properties related to every family of parts improves their definition. The composition of a family of parts becomes clearer through the properties that are related; the meaning of a property is explained by the family of parts defining its field of application.

By defining families of parts through a hierarchy with inheritance, the definition of each property can be factored to the highest family level for which this property is meaningful and may be applicable in some sub-families.

5.2 Applicable and visible properties

The properties defined at each (generic) family level are divided into two categories:

- the applicable properties corresponding to a characteristic property that every part (belonging to the generic or simple family of parts) shall possess;
- the visible properties corresponding to a characteristic property that the parts (belonging to the generic or simple family of parts) may or may not possess according to the sub-families they belong to.

Only those properties that are specified as visible for some family (possibly through inheritance) may be also specified as applicable.

5.3 Purpose of the standardised identification hierarchy

Due to the diversity of the real world, the families of parts and the properties forming the standardised identification hierarchy are not intended to exhaustively characterise and describe every part. It is only aimed at identifying and progressively characterising the common properties to provide:

- multi-supplier search;
- partial multi-supplier characterisation;
- some interchangeability between different parts.

5.4 Use of the standardised identification hierarchy

When describing his own parts a supplier may choose:

- to make reference to standardised families of parts and properties, when they are defined by the standardised identification hierarchy;
- to add his own families of parts and properties following guidance of the rules stated in this standard.

Thus every supplier can connect his families of parts to the level(s) of the standardised identification hierarchy that seem to be the most suitable and may complete the standardised identification hierarchy with his own hierarchy. The higher the level of the connection, the less the suppliers will be able to take advantage of the standardised identification hierarchy and the corresponding properties.

5.5 Class valued property

Due to the diversity of the perspectives that may be adopted for a set of objects of the real world, different structures for a generic family of parts may emerge (see example in 6.1.5).

In this case, the structure that permits the maximum applicability of the factored properties shall be selected for the standardised identification hierarchy. Each other perspective considered relevant is represented as a property that is assigned one single value to each simple family of the subtree and is valid for the whole family. The assignment of the property to a family may be defined at any level of the subtree. The value assigned this way is inherited by all the subfamilies.

The data type of such a property is represented as a set of unique codes. Each code is associated with different human readable and translatable representations.

5.6 Extensions of the common dictionary schema addressing IEC 61360 needs

The common dictionary schema combines the requirements of IEC 61360 and ISO 13584. Therefore it contains resources to accommodate the specific requirements of both International Standards. These resources are provided either as an optional capability or as subtypes of the types defined to fulfil the common requirements. The IEC 61360 specific extensions shall not be used for the definition of the standardised identification hierarchy conforming to this part of ISO 13584. These extensions are as follows:

- a) In the **item_class** entity the optional **value_code** attribute;
- b) In the **value_domain** entity the optional **terms** attribute.

5.7 Extension of the common dictionary schema addressing ISO 13584 needs

This part of ISO 13584 addresses two requirements:

- the definition of a methodology for defining standardised identification hierarchies or supplier hierarchies of parts and the properties associated with these parts.
- the specification of an information model that provides for the exchange of dictionary data that describes hierarchies of families of parts and properties of these parts.

Some resources in the common dictionary schema are intended to be used when modelling and exchanging other types of data (e.g., library content) as specified by other parts of ISO 13584. These resources are provided either as optional capabilities or as subtypes of the types defined to fulfil the requirements of this part of ISO 13584.

These extensions shall not be used when defining the standardised identification hierarchy or supplier hierarchies. These resources are as follows (the specification of these resources is defined in the informative Annex E) :

- a) **content_item**;
- b) **supplier_related_BSU**;
- c) **class_related_BSU**;
- d) **supplier_BSU_relationship**;
- e) **class_BSU_relationship**.

5.8 Compatibility between IEC 61360 and ISO 13584

The scope of IEC 61360 is to specify standard data element types with associated classification schema for electric components. This is done through the specification of the principles in IEC 61360-1, the common dictionary schema in IEC 61360-2, the set of data elements and an initial hierarchy for electronic components in IEC 61360-4, and through the specification of a Maintenance and Validation Agency intended to up-date this set of data elements and its hierarchy, to be published as IEC 61360-4.

The scope of ISO 13584 is twofold:

- to specify and to exchange identification hierarchies of parts of any kind;
- to specify and to exchange libraries of parts that may contain both dictionary data and class extensions (i. e., the set of allowed parts within a family of parts).

The exchange context of identification hierarchies is specified, for both Standards, by the common dictionary schema. Therefore exchange contexts of identification hierarchies are completely compatible. The only restriction is that ISO 13584 conforming hierarchies shall not use the two optional resources specified in 5.6 and that the date of original definition, date of current version and date of current revision that are defined as OPTIONAL in the common dictionary schema are mandatory for ISO 13584 conforming hierarchies. Moreover, the use of the ISO/IEC ICS hierarchy as a root hierarchy in the user system (see 5.1.2) enables the integration of the IEC 61360-4 hierarchy² and of the various standardised identification hierarchies intended to be developed by the ISO Standardisation Committees.

For the exchange of electric component libraries referring to the IEC 61360-4 dictionary data, the electric component suppliers can use the exchange format specified in ISO 13584-24 (in preparation). This exchange format enables, through the **component_case_of** entity, to refer to any standardised identification hierarchy conforming to the common dictionary schema.

6 Rules for creating hierarchies of families of parts

To simplify the analysis when creating a standardised identification hierarchy and supplier hierarchies, the following rules, that allow the splitting of the analysis into two phases, shall be applied. The first set of rules refers to the choice of the family hierarchy. The second set of rules refers to the association of properties with families of parts.

6.1 Choice of family hierarchy

6.1.1 Field of application

RULE 1 - Field covered by the hierarchy.

The standardised identification hierarchy shall be able to cover all the parts dealt with in ISO or IEC standards.

A part supplier hierarchy shall be able to cover all the parts supplied by this supplier.

6.1.2 Upper section of the hierarchy

RULE 2 - Classification.

The upper section of the standardised identification hierarchy shall be based upon the International Classification of Standards (ICS).

Properties can be related to the upper section according to the factoring rule (RULE 8).

NOTE 1 The ICS provides a common high level classification that shall be used or referred to by the standardisation committees through at least the root of the classification that corresponds to their scope. For those standardisation committees whose scope cannot be mapped onto one class of the ICS classification, new classes shall be created as subclasses of an existing class that encompasses their scope.

NOTE 2 When the scope of a standardisation committee encompasses a whole subtree of the ICS classification, this committee may redefine the classification subtree for its own purpose provided that it refers to the root of that subtree.

NOTE 3 The role of the standardised identification hierarchy is to improve the definition of properties. It is not intended to specify a selection mechanism.

² The IEC 1360-4 hierarchy covering several subclasses of the ISO/IEC ICS hierarchy, the former shall be connected directly to the root of the latter

6.1.3 Lower section of the hierarchy

RULE 3 - Instanciation rule.

Below the upper section of the hierarchy, a generic family of parts shall be created only when it is possible and advisable to associate a functional model (e.g. geometrical model, schematic model) to such a family of parts, i.e. when an user could reasonably choose a generic part of such a family to represent a significant state (phase) of his design process.

EXAMPLE To group screws for metal, wood and sheet metal within one family "screws", is contradictory to the rule of instanciation. During a design process a designer would never select a screw without knowing if this screw shall be a machine screw or a wood screw. Therefore, it is necessary to define below the upper section "bolts/screws/studs" different generic families of parts for the different types of screws.

6.1.4 Simple family of parts

In paper documents, standards or catalogues, sets of parts are sometimes grouped even if the sets of properties applicable to each part of the group are not exactly the same. This shall be prohibited for a simple family of parts.

RULE 4 - Homogeneity of simple family of parts.

Every property applicable to a simple family of parts shall be applicable for every part within this family.

6.1.5 Multiple perspectives on the hierarchy

It may appear that at some point in the hierarchy several properties of a family of parts can be used to define a substructure for that hierarchy.

RULE 5a - Maximum applicability

When the application of the instanciation rule (RULE 3) leads to different possible structures of some generic families depending upon a point of view, the structure that enables the maximum applicability of the factored properties shall be selected.

RULE 5b - Class valued properties

The other points of views considered essential for selection purposes – not selected however for the structure according to the maximum applicability rule (RULE 5a) – shall be represented by means of class valued properties (see 5.5). Those are intended to be queried by the user to select a set of families that correspond to the other points of view.

RULE 5c - Class valued assignment level

A class valued property shall be assigned at the level of a family where that property applies for *every* simple family of the corresponding subtree and does not apply for the immediate parent family.

EXAMPLE The family of circular bearings may split up into ball, needle and conical bearings or into sealed bearings and non-sealed bearings. The first perspective shall be selected for structuring purposes (RULE 5a). A class valued property, named "is_sealed", that is intended to take a constant BOOLEAN value in all the lower simple families shall be visible at the circular bearing family level (RULE 5b). That property shall be assigned at the circular bearing family level if the "is_sealed" property does not apply to bearings that are non-circular bearings. That property shall be assigned at least at the bearings level if the "is_sealed" property applies to any kind of bearings (RULE 5c).

6.2 Association of properties

6.2.1 Properties to be considered

RULE 6 - Choice of properties.

As a minimum, those properties that characterise a generic family of parts and can be used when searching parts for every sub-family of this generic family of parts, shall be associated with the standardised identification hierarchy.

Those properties that are not used (or rarely used) for searching purposes can be added as appropriate.

6.2.2 Semantic identification of properties

The following rule gives two criteria for deciding when two properties have the same semantics.

RULE 7 - Semantic identification

Two characteristic properties of two different parts shall be factored to a higher level of the hierarchy, if and only if, they are considered to have the same semantic meaning. Such a decision shall be justified by satisfying one of the two following criteria:

- interchangeability criterion: the two parts can, in some circumstances, be interchangeable and when interchanged the two characteristic properties shall have identical values;
- criterion of homogeneity in processing: the two properties play an identical role with respect to some process, automatic or non automatic, that might be performed over a set of parts.

Otherwise, two different properties shall be defined.

EXAMPLE 1 The properties thread diameter of hexagon head screws and cylindrical head screws fit the interchangeability criterion.

EXAMPLE 2 The properties mass (calculation of the mass of a part or its inertia), designation (for part lists), diameter of conical attachment of tools for automatic tool exchange or the outside diameter of cylindrical electronic components for automatic placement fit the criterion of homogeneity in automatic processing.

EXAMPLE 3 A thread diameter of a screw for wood or of a screw for metal are characteristic properties which shall not be the same property.

6.2.3 Factoring rule

It is not possible in every case to define a hierarchy in which:

- every property having the same semantics in several sub-families is factored (i.e. defined only once and inherited) as an unique property at the level of the ascendant family;
- any property defined at the level of a generic family of parts applies actually (i.e. has a value) to every part in every sub-family.

To create an intermediate family in order to factorise a property may be prevented by the instantiation rule (RULE 3) that is used for defining the family hierarchy.

The following rule provides guidelines for systematically factoring of properties having the same semantics while maintaining the instantiation rule.

RULE 8 - Applicability of inherited properties.

Two properties having the same semantics (RULE 7) in two families shall be factored as an unique property defined at the level of the common ascendant family. If this property does not apply to some sub-families, it shall be defined as a visible property. This property may be specified as applicable in the various sub-families where it is visible. The definition of this property shall be such that for any sub-family there is no doubt about its applicability, and when it applies, there is no doubt about the characteristic property to which it corresponds.

EXAMPLE 1 The property "material", the definition of which would be "material from which the part is made" cannot be factored; we do not know for example to which property it corresponds for a turning tool with carbide brazed tip.

EXAMPLE 2 The property "single material", whose definition would be "property only applicable to parts made from one single material; the value of this property is the code of the single material", corresponds with the semantic identification rule (RULE 7).

7 Dictionary elements that describe properties of parts

Following the conceptual model defined in ISO 13584-10, the properties of a part are classified into:

- part characteristics
- context dependent characteristics
- context parameters

7.1 Mapping of properties onto the common dictionary schema

A property is mapped in the common dictionary schema onto a **property_BSU** that carries its identification and a **property_DET** that provides its description. A part characteristic is mapped onto the **non_dependent_P_DET** subtype (of **property_DET**), a context dependent characteristic is mapped onto the **dependent_P_DET** subtype (of **property_DET**) and a context parameter is mapped onto the **condition_DET** subtype (of **property_DET**). **NOTE** Context parameters and context dependent characteristic shall be represented at least when they prove useful for selection purposes.

The family that defines the visibility of a property is mapped onto the **property_BSU.name_scope** attribute. The applicability of a property is defined by the class(es) to which it is applicable through the **class.described_by** attribute.

In this section, only the attributes linked directly to a property are described.

7.2 Attributes

Each part property is described by a dictionary element that contains the following attributes:

- Code
- Definition Class
- Data Type
- Preferred Name
- Short Name
- Preferred Letter Symbol

Synonymous Letter Symbols

Synonymous Name

Property Type Classification

Definition

Source Document of Definition

Note

Remark

Unit

Condition

Formula

Value Format

Date of Original Definition

Date of current version

Date of current revision

Version Number

Revision Number

The entry for each attribute is structured in the following way (entries may be omitted if not applicable):

- Name of the attribute
- Obj: Objective.
- Descr: Description.
- Oblig: Obligation.
- Trans: Need for translation.
- For: Representation format or maximal number and type of characters if it is a string.
- Mapp: Mapping onto the attributes used in the common dictionary schema.
- Expl: Example.

NOTE For internal purposes within a company, other attributes and/or other codes for some attributes can be defined but they cannot be used for exchange purposes.

7.2.1 Code

- Obj: To identify a property within a family of parts and to allow an absolute identification within the data dictionary when it is associated with the code of the family of parts where the property is defined, and with the library data supplier code.

Descr: A Basic Semantic Unit associated with the property.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DETdictionary_element.identified_bybasic_semantic_unit.code**

7.2.2 Definition Class

Obj: To specify for which family the property is defined.

Descr: The code of the family that is the root of the tree where this property is visible.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DETdictionary_element.identified_byproperty_BSU.name_scope**

7.2.3 Data Type

Obj: To specify the data type of the property. The data type describes the set of values that may be assigned to a property.

Descr: A type conforming to the type system specified in the common dictionary schema that specifies the data type of the property.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DET.domain**

NOTE **property_DET.domain** refers to **data_type** which again will be subtyped for the various possible data types. Thus, the data type is directly attached to the property. However, an identification mechanism using a **named_type** and a **data_type_BSU** is also available for the cases where the same data type is intended to be used for different properties. When a **data_type** is intended to be used by different properties, the definition of this **data_type** as a **named_type** provides for separate up-dating of the data-type and of the properties that refer to it.

7.2.4 Preferred Name

Obj: To give a name of the property (in full length where possible). It is used for communication and understanding.

Descr: The preferred name is identical to the name that is used in International Standards if available. If the preferred name in the International Standard is longer than the maximum length allowed for this attribute it shall be meaningfully abbreviated.

Oblig: Mandatory.

Trans: To be translated.

For: 30, alphanumeric.

Mapp: **property_DETclass_and_property_elements.nameslitem_names.preferred_name**

Expl: Threaded diameter

7.2.5 Short Name

Obj:	To give a name of the property for representation in limited space.
Descr:	It is a meaningful abbreviation of the preferred name. If standardised abbreviations exist they should be used. It can be identical to the preferred name or letter symbol.
Oblig:	Mandatory.
Trans:	To be translated, in the case that the short name is identical to the letter symbol, it shall be the same in all languages.
For:	15, alphanumeric.
Mapp:	property_DET.class_and_property_elements.names.litem_names.short_name
Expl:	thread_diam.

7.2.6 Preferred Letter Symbol

Obj:	To allow a shorter name of the property. When it exists, it is used in place of the short name for representation in tables, formula, drawings etc...
Descr:	The letter symbol is unique within a simple family of parts. It shall be derived from International Standards (e. g. ISO 31, IEC 27, IEC 148 and product standards). The preferred letter symbol is always provided in a text representation. It may also be provided in an SGML ISO 8879 representation.
Oblig:	Optional.
Trans:	No translation.
For:	Alphanumeric and SGML format (if provided).
Mapp:	property_DET.preferred_symbol

7.2.7 Synonymous Letter Symbol

Obj:	To allow further shorter name of the property. It is used for representation in tables, formula, drawings etc...
Descr:	The letter symbol is unique within a simple family of parts. It shall be derived from International Standards (e.g. ISO 31, IEC 27, IEC 148 and product standards). None, one or more synonym letter symbols are allowed. The synonym letter symbol is always provided in a text representation. It may also be provided in an SGML representation.
Oblig:	Optional.
Trans:	No translation.
For:	Alphanumeric and SGML format (if provided).
Mapp:	property_DET.synonymous_symbols

7.2.8 Synonymous Name

Obj:	Synonyms to the preferred name may be provided to facilitate transition from names used for local or historical reasons.
------	--

Descr: Alternative designation that differs from the given preferred name but represents the same concept. None, one or more synonymous names are allowed.

Oblig: Optional.

Trans: To be translated.

For: 30, alphanumeric.

Mapp: **property_DET\class_and_property_elements.names\item_names
.synonymous_names**

7.2.9 Property Type Classification

Obj: To classify the different properties defined in order to make large collections of property definitions more manageable.

Descr: The whole set of properties are divided into subsets according to the categories defined in ISO 31. The property type classification attribute is the reference to the ISO 31 category that is relevant for the property.

Oblig: Mandatory.

Trans: No translation.

For: One capital letter and two digits.

Mapp: **property_DET.DET_Classification**

NOTE 1 A survey of the main classes and categories of properties in ISO 31 is given in annex E. The ISO 31 classification of quantitative measure is given in Annex F. The ISO 31 classification of non quantitative properties, also called identifications and indicators, is given in Annex A.

NOTE 2 In IEC 61360, the term unit_of_measure is used to denote this attribute.

7.2.10 Definition

Obj: To describe the meaning of the property.

Descr: Statement that describes the meaning of the property and permits its differentiation from all other properties. It shall be a definition in the sense that it shall be complete and unambiguous. All significant words are free from homonym and synonymy.

Oblig: Mandatory.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **property_DET\class_and_property_elements.definition**

7.2.11 Source Document of Definition

Obj: A reference to the source document from which the property definition was derived.

Descr: As a minimum the reference shall be given with the document number and date of issue of the document.

Oblig: Optional.

Trans: No translation.

For: Alphanumeric. An identifier of the document.

Mapp: **property_DET\class_and_property_elements.source_doc_of_definition\identified_document.document_identifier**

NOTE In an exchange conforming to ISO CD 13584 part 24, the document itself may be exchanged.

7.2.12 Note

Obj: To provide further information on any part of the terminological record, that is essential to the understanding of that record.

Descr: It shall be copied from the definition in the source document into the definition of the property.

Oblig: Optional.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **property_DET\class_and_property_elements.note**

7.2.13 Remark

Obj: Explanatory text to further clarify the meaning of the usage of the property.

Descr: Free text remarks. It shall not influence the meaning.

Oblig: Optional.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **property_DET\class_and_property_elements.remark**

7.2.14 Unit

Obj: Prescription of the unit in which the value of a quantitative property is expressed.

Descr: Only the symbols of basic SI units shall be used.

Oblig: Mandatory (for quantitative data).

Trans: No translation.

For: The unit is represented as specified in ISO 10303-41 using, if required, the extensions specified in the common dictionary schema. A mathematical string may be provided. A mathematical string is always provided in a text representation. It may also be provided in an SGML representation.

Mapp: **property_DET.domain\int_measure_type.unit.structured_representation** or **property_DET.domain\real_measure_type.unit.structured_representation**

NOTE In IEC 61360, the term **unit_of_measure** is used to denote this attribute.

7.2.15 Condition

Obj:	To formally identify the context parameter on which a context dependent characteristic depends.
Descr:	To give the code(s) of the adequate context parameter(s).
Oblig:	Mandatory for context dependent characteristics.
Trans:	No translation.
For:	A reference to a set of Basic Semantic Units.
Mapp:	property_DETdependent_P_DET.depends_on

7.2.16 Formula

Obj:	Rule or statement in mathematical form expressing semantics of a quantitative property.
A	formula shall not change any essential information of the meaning of that definition.
Descr:	It is a mathematical expression of the property definition.
Oblig:	Optional.
Trans:	No translation.
For:	A mathematical string (a mathematical string is always provided in a text representation; it may also be provided in an SGML representation).
Mapp:	property_DET.formula

7.2.17 Value Format

Obj:	Specification of the type and length of the representation of the value of a property. It is intended as a maximum <u>value format</u> for communication and database storage.
Descr:	The value format shall be defined according to the definition below: a. Non-quantitative data value format types [ISO 9735: 1988]: - A = alphabetic characters, letters only - M = mixed, all characters allowed - N = numeric characters, digits only - X = alphanumeric characters, letters or digits only. b. Quantitative data value format types [ISO 6093: 1985]: - NR1 = integers, - NR2 = rational numbers with decimal-mark (real), - NR3 = rational numbers with decimal-mark and exponent-mark (floating point),

S^3 = signed (positive or negative),

. = decimal-mark,

E = exponent-mark, base 10: (A)E(B) represents the value $A \times 10^B$

c. Field length

The field length of a non-quantitative data value shall be indicated by a number (e.g. 17).

The following preferred standard formats -derived from ISO 9735 and ISO 6093 have been defined:

A..3 N..3 X..3 M..3

A..8 N..8 X..8 M..8

A..17 N..17 X..17 M..17

A..35 N..35 X..35 M..35

A..(nx35) N..(nx35) X..(nx35) M..(nx35)

In these formats no special characters shall be allowed. A variable field length shall start with two dots. A fixed field length shall start with one space (examples: A 3 - N 8 - X 17 - M 35 etc.).

The field length of a quantitative data value shall be indicated by a combination of digits and characters (e.g. 3.3ES2). The following preferred standard formats -derived from ISO 9735 and ISO 6093- have been defined:

NR1..4 positive integers

NR1 S..4 positive or negative integers

NR2..3.3 positive reals

NR2 S..3.3 positive or negative reals

NR3..3.3ES2 floating point, positive

NR3 S..3.3ES2 floating point, positive or negative

In these formats no special characters shall be allowed. A variable field length shall start with two dots. A fixed field length shall start with one space (examples: NR1 4 - NR1 S 4 etc.).

Oblig: Mandatory.

Trans: No translation.

For: 15, alphanumeric.

Mapp: **property_DET.domain\simple_type.value_format**

NOTE In case of named types the mapping may be done through a chain of **named_types**.

3 In addition to ISO 6093 the sign ' S ' is used as a marker for ' signed ' (should read ' Sign ').

7.2.18 Date of Original Definition

Obj:	To show when the property was defined by the library data supplier and thus when it was declared as valid by this supplier. This date will never be changed and can be used for verification purpose.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	property_DET\dictionary_element.time_stamps\dates.date_of_original_definition
Expl:	1967-08-20.

7.2.19 Date of Current Version

Obj:	To show the date when the current version was defined.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	property_DET\dictionary_element.time_stamps\dates.date_of_current_version

7.2.20 Date of Current Revision

Obj:	To show the date of the last revision number change.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	property_DET\dictionary_element.time_stamps\dates.date_of_current_revision

7.2.21 Version Number

Obj:	To characterise each version of a property. A new version number of a property shall be created whenever a change in some attribute that describes this property influences its use.
NOTE 1	No change is allowed that affects the meaning of a property.
NOTE 2	The changes in a property that affect its version number are defined in 7.3.1, Table 1.
Descr:	A string that contains a natural number to indicate the different versions of a property during the life cycle. Consecutive version numbers shall be issued in ascending order. A new version of the property shall be generated according to the definitions given in table 1 (see 7.3).

Oblig: Mandatory.

For: 3, numerical.

Trans: No translation.

Mapp: **property_DETdictionary_element.identified_bybasic_semantic_unit.version**

7.2.22 Revision Number

Obj: To characterise each revision of the same version of a property. A new revision number of a property shall be created when a change in some attribute that describes this property influence neither its meaning nor its use.

NOTE 1 No change is allowed that affects the meaning of a property.

NOTE 2 The changes in a property that affect its revision number are defined in 7.3.1, table 1.

Descr: A string that contains a natural number used for administrative control of a property. Consecutive revision numbers shall be issued in ascending order for each value of the version of a property. Per property, unique by its identifier, only one revision number is current at any time. A new revision number of the property shall be generated according to the definitions given in table 1 (see 7.3).

Oblig: Mandatory.

For: 3, numerical.

Trans: No translation.

Mapp: **property_DETdictionary_element.revision**

7.3 Rules for defining new versions and/or revision of properties

7.3.1 Changes in the attribute of properties

Two concepts are provided to mark updating operations for properties. The concept of version number is a tag for identifying data (it is related to a BSU) whereas the concept of revision number is one for description data of properties (it is related to a Dictionary Element).

The following table (see table 1) gives an overview of how updating operations (add, modify, delete of each attribute of a property) have effects on the version number (**V**) or the revision number (**R**) concept for the given property. There are cases where the denoted operation is meaningless on the conceptual level, and syntactically constrained in the common dictionary schema (–), or forbidden at all (**X**). The table does not reflect the creation of a property. To modify means to modify the value of the attribute.

Table 1 — Overview of versioning in property updating operations

Attribute	Add	Modify	Delete
Code	–	X	X
Definition Class	X	X	X
Data type	–	V ⁴	X
Preferred Name	–	R	X
Short Name	–	R	X
Preferred Letter Symbol	R	R	X
Synonymous Letter Symbol	R	R	R
Synonymous Name	R	R	R
Property Type Classification	–	R ⁵	X
Definition	–	R/V ⁶	X
Source Document of Definition	R ⁷	R ⁸	R
Note	R ⁸	R ⁸	R
Remark	R ⁸	R ⁵	R
Unit	–	X	X
Condition	X	X/R ⁸	X
Formula	R ⁸	R ⁸	R ⁸
Format	–	V	X
Date of Original Definition	–	X	X
Date of current version	–	V	X
Date of Current Revision	–	R / V	X
Version Number	–	V	X
Revision Number	–	R	X

7.3.2 Propagated changes in property version numbers

The following table (see table 2) specifies which changes in a property attribute shall lead to new version of this property.

4 The SUBTYPE of the **data_type** entity that defines the domain can not be changed (X). The only allowed changes consist of

1 - Adding new allowed values in an enumerated type, modelled either as a **non_quantitative_code_type** entity or a **non_quantitative_int_type** entity; or

2 - Changing the version of the class that constitutes the domain of the property (is_part_of relationship for assembly modelling).

3 - Changing the version of the **data_type_BSU** that identifies the domain of the property

These three changes modify the version (V) of the property.

5 Only allowed in case of revision of ISO 31 or of this part of the ISO 13584 series.

6 The changes in definition shall not affect the abstract property referred to by the definition. Nevertheless it may change the specification of the measure process to be used to evaluate the value of this property. If the changes in the definition affect this measure process it leads to a new version, else it leads to a new revision.

7 This change shall not affect the abstract property referred to by the property attributes.

8 The context parameters (modelled as **condition_DET**) on which a context dependant characteristic modelled as **dependent_P_DET** depend shall not be changed. Nevertheless, change in the version of some of these **condition_DET**s leads to change of version of the **dependent_P_DET**.

Table 2 — Propagated changes in version numbers of properties

Attribute	Changed item
Data type	<code>property_DET.domain\non_quantitative_code_type.domain.its_values⁹</code> <code>property_DET.domain\non_quantitative_int_type.domain.its_values⁶</code> <code>property_DET.domain\Class_Instance_Type.domain.version</code> <code>property_DET.domain\Named_Type.referred_type.version</code>
Condition	<code>property_DET\dependent_P_DET.depends_on[.].version¹⁰</code>

8 Dictionary elements that describe families of parts

8.1 Mapping of families onto the common dictionary schema

A family is mapped in the common dictionary schema onto a **class_BSU** that carries its identification and a **component_class** that provides its description.

In this section, only the attributes linked directly to a class are described.

8.2 Attributes

Each family of parts is described by a dictionary element that contains the following attributes:

- Code
- Superclass
- Preferred Name
- Short Name
- Synonymous Name
- Visible Types
- Applicable Types
- Sub-class Selection Properties
- Visible Properties
- Applicable Properties
- Class Value Assignment
- Definition,
- Source Document of Definition

⁹ Values shall only be added

¹⁰ Version change of any property in the set

Note

Remark

Simplified Drawing

Date of Original Definition

Date of Current Version

Date of Current Revision

Version Number

Revision Number

The entry for each attribute is structured in the following way (entries may be omitted if not applicable):

- Name of the attribute;
- Obj: Objective;
- Descr: Description;
- Oblig: Obligation;
- Trans: Need for translation;
- For: Representation format or maximal number and type of characters if it is a string;
- Mapp: Mapping onto the resource constructs used in the common dictionary schema;
- Expl: Example.

8.2.1 Code

- Obj: To identify a family of parts and to allow an absolute identification within the data dictionary when associated with the library data supplier code.
- Descr: A Basic Semantic Unit associated with the family of parts.
- Oblig: Mandatory.
- Trans: No translation.
- Mapp: **component_classdictionary_element.identified_by/basic_semantic_unit.code**

8.2.2 Superclass

- Obj: To reference the immediate (unique) parent family of a family.
- Descr: A Basic Semantic Unit of the immediate parent family of the current family.
- Oblig: Optional (if it does not exist, the class has no superclass).
- Trans: No translation.
- For: A Class Basic Semantic Unit.

Mapp: **component_classclass.its_superclass**

NOTE 1 When defining a standardised identification hierarchy, all the defined classes shall have a superclass. The root of such a tree shall be either one class of the ISO/IEC International Classification for Standards (ICS) predefined tree, or a class already defined by another standardised identification hierarchy.

NOTE 2 The code of the class that correspond to each node of the tree defined in the ISO/IEC ICS shall be equal to the code defined in ICS: 1993, where dots are replaced by underscores. The code of the ICS: 1993 root shall be 'OO'.

NOTE 3 The version of the class that corresponds to each node of the tree defined in the ISO/IEC ICS shall be equal to '001', unless other versions be specified by an ISO/IEC standard that integrates, in its normative references, both ISO 13584-42 and IEC 61360-2.

NOTE 4 The **Class_BSU**s that carry the identity of the classes that correspond to the node of the tree defined in the ISO/IEC ICS shall refer in their **defined_by** attribute, to the special code defined in ISO 13584-26 for ISO/IEC ICS..

8.2.3 Preferred Name

Obj: To give a meaningful description of the family of parts (in full length where possible). It is used for communication and understanding.

Descr: The preferred name is identical to the name that is used in International Standards if available. If the preferred name in the International Standards is longer than the max. length allowed for this attribute it shall be meaningfully abbreviated.

Oblig: Mandatory.

Trans: To be translated.

For: 30, alphanumeric.

Mapp: **component_classclass_and_property_elements.nameslitem_names.preferred_name**

Expl: Screw threads.

8.2.4 Short Name

Obj: To give a name to the family of parts for representation in limited space.

Descr: It is a meaningful abbreviation of the preferred name. If standardised abbreviations exist they should be used. It can be identical to the preferred name.

Oblig: Mandatory.

Trans: To be translated, in the case that the short name is identical to the letter symbol, it will be the same in all languages.

For: 15, alphanumeric.

Mapp: **component_classclass_and_property_elements.nameslitem_names.short_name**

8.2.5 Synonymous name

Obj: Synonyms to the preferred name may be provided to facilitate transition from names used for local or historical reasons.

Descr: Alternative designation that defers given preferred name but represents the same concept. None, one or more synonymous names are allowed.

Oblig:	Optional.
Trans:	To be translated.
For:	30, alphanumeric.
Mapp:	component_class class_and_property_elements.names litem_names.synonymous_names

8.2.6 Visible Types

Obj:	To define the new named types that may be referred to, as their data type, by the visible properties of this family, or of any of its subfamilies (visible type).
Descr:	Different properties may have the same data type of values (e.g., a set of codes that identify materials). Such data types may be defined as named types, independently of any property. They may be, later on, referred to as their data type by different properties. The definition of named type shall contain the code of the named type, the version number of the named type, and the data type of the named type specified by using the resource constructs of the common dictionary schema. For the non_quantitative_code_type and non_quantitative_int_type the data type shall be specified as a set of dic_values each one consisting of an unique code and a set of names (possibly translated).
Oblig:	Optional.
Trans:	In case of non_quantitative_code_type and non_quantitative_int_type value names are to be translated
For:	data_type_BSUs and data_type_elements
Mapp:	component_class dictionary_element.identified_by lclass_BSU.added_visible_data_types

NOTE 1 Visible types are inherited.

NOTE 2 Visible types that are not also applicable types may only be referenced to define the data type of the visible properties of the family (or of any of its subfamilies).

EXAMPLE A **named_type** "material" may be defined at the level of some family as a **non_quantitative_code_type**. The set of values of this named type consists of a set of codes associated with different (translated) names. This named type may be referenced in some (sub-) family to define the data type of the property that corresponds to the material the parts of the family consist of. It may be referenced in some other (sub-) family to define the data type of the property that corresponds to the coating of the parts of the family.

8.2.7 Applicable Types

Obj:	To define which visible types are allowed as data type for the properties applicable to the family (or any of its sub-families).
Descr:	The list of the code of the defined or inherited visible types of the family that become applicable types for the family (and any of its sub-families).
Oblig:	Optional.
Trans:	No translation.
For:	A list of data_type_BSUs .

Mapp: **component_class\class.defined_types.**

8.2.8 Sub-class Selection Properties

Obj: To define which (new) class valued properties shall be assigned a value in any simple part family that is a sub-family of the current family.

Descr: A class valued property enables representation of different perspectives on the set of parts that belongs to the current family. When a class valued property is defined at the level of some family, the dictionary user may query all the simple sub-families of this family for which this class valued property is assigned some value.

Oblig: Optional.

Trans: No translation.

For: A list of **property_BSUs**.

Mapp: **component_class\item_class.sub_class_properties**

NOTE 1 Sub-class selection properties are inherited. They shall not appear in the sub-class selection properties list of any sub-family.

NOTE 2 In each simple family of parts, each class valued property defined as sub-class selection properties in any upper family shall be assigned a value.

8.2.9 Visible Properties

Obj: To specify the new properties that are defined at the level of this family and may be thus specified as applicable to the family or any of its sub-families (visible properties).

Descr: The new properties (with respect to inheritance) that the parts belonging to the generic or simple family of parts may or may not possess according to the subfamilies they belong to.

For: A set of **property_BSUs**.

Oblig: Mandatory (possibly empty).

Mapp: **component_class\dictionary_element.identified_by\class_BSU.added_visible_properties**

8.2.10 Applicable Properties

Obj: To specify the new properties that are specified as applicable for that family and for any of its sub-families.

Descr: The new properties (with respect to inheritance) that the parts belonging to the generic or simple family of parts shall possess. The properties of this list shall be visible for the family, i.e. they shall be defined as visible either by this family or by a family higher in the hierarchy. The LIST order shall correspond to the default presentation order of the properties in any case where such an order shall be defined (e.g., to display the properties of some classes on a screen).

For: List of **property_BSUs**.

Oblig: Mandatory (possibly empty).

Mapp: **component_class\class.described_by**

8.2.11 Class Value Assignment

- Obj: To define the values assigned to some class valued properties specified as Sub-Class Selection Properties in the family or any family higher in the hierarchy.
- Descr: Class valued properties are intended to be queried by the user to select a set of families for which such a property has a given value. Once a value is assigned to a class valued property, this value is inherited and shall not be redefined.
- For: A set of **Class_Value_Assignments**.
- Oblig: Optional.
- Mapp: **component_classlitem_class.class_constant_values**

8.2.12 Definition

- Obj: To clarify the meaning of the family by giving its intention textually.
- Descr: Statement that describes the meaning of the family and permits its differentiation from all other families. It shall be a definition in the sense that it shall be complete and unambiguous. All significant words are free from homonymy (homonym) and synonymy.
- Oblig: Mandatory.
- Trans: To be translated.
- For: Unlimited alphanumeric string.
- Mapp: **component_classclass_and_property_elements.definition**

8.2.13 Source Document of Definition

- Obj: A reference to the source document from which the family definition was derived.
- Descr: As a minimum the reference shall be given with the document number and date of issue of the document.
- Oblig: Optional.
- Trans: No translation.
- For: Alphanumeric. An identifier of the document.
- Mapp: **component_classclass_and_property_elements
.source_doc_of_definition\identified_document.document_identifier**

NOTE In an exchange conforming to ISO CD 13584 part 24, the document itself may be exchanged.

8.2.14 Note

- Obj: To provide further information on any part of the terminological record, that is essential to the understanding of that record.
- Descr: It shall be copied from the definition in the source document into the definition of the family of parts.
- Oblig: Optional.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **component_classclass_and_property_elements.note**

8.2.15 Remark

Obj: Explanatory text to further clarify the meaning of the usage of the family of parts.

Descr: Free text remarks. It shall not influence the meaning.

Oblig: Optional.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **component_classclass_and_property_elements.remark**

8.2.16 Simplified Drawing

Obj: To provide a visualisation, on request of the user, that shows an image of the class of parts.

Descr: A drawing including at least the reference coordinate system of the part (that is to be used for all the representations of this part), and the letter symbols of the main applicable properties.

Oblig: Mandatory for all the families that may be instantiated.

Trans: No translation.

For: An ABSTRACT SUPERTYPE.

Mapp: **component_classlitem_class.simplified_drawing**

NOTE ISO CD 13584 part 24 provides an exchange format for such drawings.

8.2.17 Date of Original Definition

Obj: To show when the family of parts was defined by the library data supplier and thus when it was declared as valid by this supplier. This date will never be changed and can be used for verification purpose.

Descr: The entry shall be in accordance with ISO 8601.

Oblig: Mandatory.

Trans: No translation.

For: 10, alphanumeric.

Mapp: **component_classdictionary_element.time_stamps.date_of_original_definition**

Expl: 1967-08-20

8.2.18 Date of Current Version

Obj: To show the date when the current version was defined.

Descr: The entry shall be in accordance with ISO 8601.

Oblig: Mandatory.

Trans: No translation.

For: 10, alphanumeric.

Mapp: **component_classdictionary_element.time_stamps.date_of_current_version**

8.2.19 Date of Current Revision

Obj: To show the date of the last revision number change.

Descr: The entry shall be in accordance with ISO 8601.

Oblig: Mandatory.

Trans: No translation.

For: 10, alphanumeric.

Mapp: **component_classdictionary_element.time_stamps.date_of_current_revision**

8.2.20 Version Number

Obj: To characterise each version of a class. A new version number of a class shall be defined when ever a change in the attributes that describes this class influences its use.

NOTE 1 No change is allowed that affects the meaning of a class.

NOTE 2 The changes in a class that affect its version number are defined in 7.3.1, table 3.

Descr: A string that contains a natural number to indicate the different versions of a family during the life cycle. The version number string of a property shall consist of three numeric characters. Consecutive version numbers shall be issued in ascending order. A new version number of the property shall be generated according to the definitions given in table 3 (see 7.3).

Oblig: Mandatory.

For: 3, numeric.

Trans: No translation.

Mapp: **component_classdictionary_element.identified_by/basic_semantic_unit.version**

8.2.21 Revision Number

Obj: To characterise each revision of a version of a class. A new revision number of a class shall be defined when ever a change in the attributes that describes this class influences neither its meaning nor its use.

NOTE 1 No change is allowed that affects the meaning of a class.

NOTE 2 The changes in a class that affect its revision number are defined in 7.3.1, table 3.

Descr:	A string that contains a natural number used for administrative control of a class. Consecutive revision numbers shall be issued in ascending order for each value of the version number of a class. Per class, unique by its identifier, only one revision number is current at any time. A new revision number of the class shall be generated according to the definitions given in table 3 (see 7.3).
For:	3, numeric.
Trans:	No translation.
Mapp:	component_classdictionary_element.revision

8.3 Rules for defining new version or revision of classes

8.3.1 Change in the attribute of class

Two concepts are provided to mark updating operations for families of parts. The concept of version number is a tag for identifying data (it is related to a BSU) whereas the concept of revision number is one for description data of families of parts (it is related to a Dictionary element).

The following table gives an overview of how updating operations (add, modify, delete of some attributes of a class) have effects on the version number (**V**) or the revision number (**R**) concept for the given class. There are cases, where the denoted operation is meaningless, and syntactically constrained in the common dictionary schema (–), or forbidden at all (**X**). The tables do not reflect the creation of a family. To modify means to modify the value of the attribute.

Table 3 — Overview of versioning in class updating operations

Attribute	Add	Modify	Delete
Code	–	X	X
Superclass	V ¹¹	V ⁸	V ⁸
Preferred Name	–	R	X
Short Name	–	R	X
Visible Types	V	V ¹²	X
Applicable Types	V	V ⁹	X
Sub-Class Selection Properties	V	V ⁹	X
Synonymous name	R	R	R
Visible Properties	V	V ⁹	X
Applicable Properties	V	V ⁹	X
Class Value Assignment	V	V ⁹	X
Definition	–	R ¹³	X
Source Document of Definition	R	R	R
Note	R	R	R
Remark	R	R	R
Simplified Drawing	–	R	R
Date of Original Definition	–	X	X
Date of current version	–	V	X
Date of Current Revision	–	R / V	X
Version number	–	V	X
Revision number	–	R	X

8.3.2 Propagated changes in family version numbers

Any change of version number of a family leads to a change of version number of its sub-family (see table 3: Modify Superclass), and of the sub-families of this family, and so on. *Therefore, all the changes that change the version number of the inherited properties or types of a family change the version number of this family.*

When a part family, intentionally defined by a set of dictionary data, belongs to a parts library, the class that constitutes the information model of this part family may also reference other basic semantic units (tables and documents), and it may be associated with a **content_item** that defines the extension of this class.

11 Modification of superclass shall not remove any inherited visible or applicable property or type. It may consist either in changing the version of the referenced superclass, that defines a new version of the subclass, or, e.g., in adding an intermediate class in between one class and its previous superclass.

12 Modification shall consist only in addition of new items or in change in version of referenced items. No other modification shall take place. No item shall be removed.

13 This change shall not affect the abstract population referred to by the class data element type.

NOTE 1 The concept of extension of a class is defined in ISO CD 13584 part 24.

NOTE 2 The information model of a family of parts in a parts library is defined in ISO 13584 -24¹⁴.

Any change in the **content_version** that characterises the extension of a family of parts in a parts library, and any change in the version number of the tables that are referenced from the class that constitutes the information model of the family of parts or in the version number of the tables that reference this class through their **name_scope** attribute shall lead to a new version number of the family.

14 To be published