



BSI Standards Publication

Industrial automation systems and integration — Parts library

Part 24: Logical resource: Logical model of supplier library

National foreword

This British Standard is the UK implementation of [ISO 13584-24:2003](#).

The UK participation in its preparation was entrusted to Technical Committee AMT/4, Industrial data and manufacturing interfaces.

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

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 22187 9

ICS 25.040.40

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

ISO
13584-24

First edition
2003-11-01

Industrial automation systems and integration — Parts library —

Part 24: Logical model of supplier library

*Systèmes d'automatisation industrielle et intégration — Bibliothèque
de composants —*

Partie 24: Ressource logique : Modèle logique de fournisseur



Reference number
ISO 13584-24:2003(E)

AVA Documents
Document Sharing Hub
© ISO 2003



COPYRIGHT PROTECTED DOCUMENT

© ISO 2003, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	xv
Introduction	xvi
1 Scope	1
2 Normative references	2
3 Terms, definitions and abbreviations	3
4 Structure of ISO 13584-24	16
4.1 Generic resources	17
4.1.1 ISO13584_instance_resource_schema	17
4.1.2 ISO13584_library_expressions_schema	17
4.1.3 ISO13584_table_resource_schema	17
4.1.4 ISO13584_variable_semantics_schema	17
4.1.5 ISO13584_domain_resource_schema	17
4.2 Parts library specific resources	18
4.2.1 ISO13584_extended_dictionary_schema	18
4.2.2 ISO13584_library_content_schema	18
4.2.3 ISO13584_external_file_schema	18
4.2.4 ISO13584_method_schema	18
4.3 Library integrated information models	18
4.3.1 ISO13584_g_m_iim_schema and LIIM 24-1	19
4.3.2 ISO13584_f_m_iim_schema and LIIM 24-2	19
4.3.3 ISO13584_f_v_iim_schema and LIIM 24-3	19
5 Fundamental concepts and assumptions	19
5.1 Conceptual model of a supplier library	19
5.2 Implicit versus explicit description of a parts library	20
5.2.1 Explicit modelling of simple families of parts: by set extension	20
5.2.2 Implicit modeling of simple families by entity data type	21
5.2.3 Explicit and implicit description of classes in this part of ISO 13584	21
5.3 Direct use of EXPRESS versus meta-modelling for implicit description	22
5.3.1 Direct use of the EXPRESS language for modelling classes	22
5.3.2 Meta-modelling of classes using EXPRESS	23
5.4 Two level description of a supplier library and the ISO/IEC common dictionary schema	23
5.4.1 Common dictionary description for ISO 13584 and IEC 61360	24
5.4.2 Dictionary descriptions for ISO 13584	24
5.4.3 Interoperability of ISO 13584 and IEC 61360	24
5.5 Independence between dictionary_elements and content_items: the BSU mechanism	25
5.5.1 Reference between several EXPRESS schema populations via the BSU mechanism	25
5.5.2 Expressing constraints between dictionary entries	25
5.6 ISO 13584 and the Internet	26
5.6.1 Documents represented within a library exchange context	26
5.6.2 Support of the HTTP protocol and local Internet server	26
5.6.3 Particular HTTP formats to be supported by an implementation	26
5.6.4 Remote access to a document through the Internet	27
6 ISO13584_instance_resource_schema	28
6.1 Introduction to the ISO13584_instance_resource_schema	29
6.2 Fundamental concepts and assumptions for the ISO13584_instance_resource_schema	29
6.2.1 Two-fold description of classes and instance representation	29
6.2.2 Representation of a context-dependent characteristic value	31
6.2.3 Optional properties	31
6.3 ISO13584_instance_resource_schema type definitions	31

6.3.1	Null_value.....	32
6.3.2	Primitive_value.....	32
6.3.3	Null_or_primitive_value.....	32
6.3.4	Simple_value.....	32
6.3.5	Null_or_simple_value.....	32
6.3.6	Number_value.....	32
6.3.7	Null_or_number_value.....	32
6.3.8	Integer_value.....	33
6.3.9	Null_or_integer_value.....	33
6.3.10	Real_value.....	33
6.3.11	Null_or_real_value.....	33
6.3.12	Boolean_value.....	33
6.3.13	Null_or_boolean_value.....	33
6.3.14	Translatable_string_value.....	33
6.3.15	Translated_string_value.....	34
6.3.16	String_value.....	34
6.3.17	Null_or_translatable_string_value.....	34
6.3.18	Complex_value.....	34
6.3.19	Null_or_complex_value.....	34
6.3.20	Entity_instance_value.....	35
6.3.21	Null_or_entity_instance_value.....	35
6.3.22	Defined_entity_instance_value.....	35
6.3.23	Controlled_entity_instance_value.....	36
6.3.24	STEP_entity_instance_value.....	36
6.3.25	PLIB_entity_instance_value.....	36
6.3.26	Uncontrolled_entity_instance_value.....	36
6.3.27	Property_or_data_type_BSU.....	36
6.4	ISO13584_instance_resource_schema entity definitions.....	36
6.4.1	Level_spec_value.....	37
6.4.2	Null_or_level_spec_value.....	37
6.4.3	Int_level_spec_value.....	37
6.4.4	Null_or_int_level_spec_value.....	37
6.4.5	Real_level_spec_value.....	37
6.4.6	Null_or_real_level_spec_value.....	38
6.4.7	Class_instances.....	38
6.4.8	Property_value.....	43
6.4.9	Context_dependent_property_value.....	44
6.5	ISO13584_instance_resource_schema rule definition.....	44
6.5.1	Valued_properties_are_allowed_for_implicit_spec_rule rule.....	44
6.5.2	Valued_properties_are_allowed_for_explicit_spec_rule rule.....	44
6.5.3	Identification_properties_are_valued_for_implicit_spec_rule rule.....	45
6.5.4	Identification_properties_are_valued_for_explicit_spec_rule rule.....	45
6.5.5	Fm_valued_properties_are_allowed_for_implicit_spec_rule rule.....	45
6.5.6	Fm_valued_properties_are_allowed_for_explicit_spec_rule rule.....	46
6.5.7	Fm_free_properties_are_valued_for_implicit_spec_rule rule.....	46
6.5.8	Fm_free_properties_are_valued_for_explicit_spec_rule rule.....	46
6.6	ISO13584_instance_resource_schema function definitions.....	47
6.6.1	Compatible_class_and_class function.....	47
6.6.2	Right_values_for_level_spec function.....	47
6.6.3	Compatible_level_type_and_instance function.....	48
6.6.4	Compatible_type_and_value function.....	48
6.6.5	Collects_assigned_instance_properties function.....	49
6.6.6	Correct_view_from_model function.....	49
6.6.7	Is_condition_det function.....	49
6.6.8	Is_dependent_p_det function.....	50
6.6.9	All_context_parameters_referenced function.....	50
6.6.10	Collects_property_context function.....	50
6.6.11	Check_class_type_for_dic_item_instance function.....	50

6.6.12	Check_class_type_for_dic_f_model_instance function	51
6.6.13	Check_class_type_for_dic_f_view_instance function	51
6.6.14	Check_property_values_translations function	51
6.6.15	Same translations function	51
6.6.16	Compatible_item_caseof_with_class_definition function	52
6.6.17	Compatible_model_caseof_with_class_definition function	52
6.6.18	superclass_closure function	52
6.6.19	compute_superclass_closure procedure	52
6.6.20	item_caseof_closure function	53
6.6.21	next_item_caseof function	53
6.6.22	compute_item_caseof_closure procedure	53
6.6.23	model_caseof_closure function	54
6.6.24	next_model_caseof function	54
6.6.25	compute_model_caseof_closure procedure	54
7	ISO13584_library_expressions_schema	54
7.1	Introduction to the ISO13584_library_expressions_schema	55
7.2	Fundamental concepts and assumptions for the ISO13584_library_expressions_schema	56
7.2.1	Information model of a variable	56
7.2.2	Strong typing of variables and expressions	56
7.3	ISO13584_library_expressions_schema type definitions	57
7.3.1	Library_expression	57
7.3.2	Library_variable	57
7.4	ISO13584_library_expressions_schema entity definitions	57
7.4.1	Level_spec_expression	58
7.4.2	Entity_instance_expression	60
7.4.3	Class_instance_expression	61
7.4.4	Exists_value	66
7.4.5	Instance_comparison_equal	66
7.5	ISO13584_library_expressions_schema rule definition	66
7.5.1	Two_fold_variable_representation_rule rule	67
7.5.2	Syntax_of function	67
7.5.3	Semantics_of function	67
7.5.4	Collects_assigned_properties function	67
7.5.5	Collects_referenced_library_expressions function	68
7.5.6	Compatible_simple_type_and_expression function	68
7.5.7	Compatible_type_and_library_expression function	68
7.5.8	Compatible_variable_and_expression function	69
7.5.9	Compatible_variable_and_library_expression function	69
8	ISO13584_table_resource_schema	69
8.1	Introduction to the ISO13584_table_resource_schema	70
8.2	Fundamental concepts and assumptions for the ISO13584_table_resource_schema	71
8.2.1	Description of tables	71
8.2.2	Description of table expressions	72
8.3	ISO13584_table_resource_schema entity definitions	72
8.3.1	Table_identification	72
8.3.2	Table_specification	73
8.3.3	Table_extension	73
8.3.4	Column	75
8.3.5	Simple_column	75
8.3.6	Complex_column	76
8.3.7	Table expressions	78
8.4	ISO13584_table_resource_schema functions definition	85
8.4.1	Compatible_column_and_variable function	85
8.4.2	Compatible_column_and_variable_semantics function	85
8.4.3	Compatible_list_variable_semantics_and_columns function	86
8.4.4	Compatible_variable_semantics_and_expression function	86

8.4.5	Compatible_list_variable_semantics_and_expressions function.....	86
8.4.6	Collects_columns function.....	86
8.4.7	Diff_columns function.....	87
8.4.8	Return_key function.....	87
8.4.9	Is_SQL_mappable_table_expression function.....	88
8.4.10	Used_table_literals function.....	88
8.4.11	Check_iterator_context function.....	88
8.4.12	Check_iterator_domain_uniqueness function.....	89
8.4.13	No_null_values_in_key_columns function.....	89
8.4.14	Same_translations_for_string_values function.....	89
8.4.15	Same_translations_for_table_extension function.....	89
8.4.16	Get_translated_string_values_of_tuple function.....	90
9	ISO13584_variable_semantics_schema.....	90
9.1	Introduction to the ISO13584_variable_semantics_schema.....	90
9.2	Fundamental concepts and assumptions for the ISO13584_variable_semantics_schema.....	91
9.2.1	Instance related operation.....	91
9.2.2	Instance structure.....	91
9.2.3	Context of a method.....	91
9.3	ISO13584_variable_semantics_schema type definition.....	91
9.3.1	Property_semantics_or_path.....	91
9.4	ISO13584_variable_semantics_schema entity definitions.....	91
9.5	Property_semantics.....	91
9.6	Sub_property_path.....	92
9.7	Variable_semantics referring to the SELF entity.....	93
9.7.1	Self_variable_semantics.....	93
9.7.2	Self_property_semantics.....	93
9.7.3	Self_property_value_semantics.....	93
9.7.4	Self_property_name_semantics.....	94
9.7.5	Self_class_variable_semantics.....	96
9.7.6	Self_class_name_semantics.....	96
9.8	Variables referring to the open view characteristics.....	98
9.8.1	Open_view_variable_semantics.....	98
9.8.2	Open_view_property_semantics.....	98
9.8.3	Open_view_property_value_semantics.....	99
9.9	ISO13584_variable_semantics_schema function definitions.....	99
9.9.1	BSU_of_property_semantics function.....	99
9.9.2	Check_property_semantics function.....	99
10	ISO13584_domain_resource_schema.....	99
10.1	Introduction to the ISO13584_domain_resource_schema.....	100
10.2	Fundamental concepts and assumption for the ISO13584_domain_resource_schema.....	101
10.3	ISO13584_domain_resource_schema type definition.....	102
10.3.1	Boolean_expression_or_others.....	102
10.4	ISO13584_domain_resource_schema entity definitions.....	102
10.4.1	Others.....	102
10.4.2	Domain_restriction.....	102
10.4.3	Guarded_simple_domain.....	103
10.4.4	Simple_domain.....	103
10.4.5	Table_defined_domain.....	103
10.4.6	Type_defined_domain.....	104
10.4.7	Subclass_defined_domain.....	104
10.4.8	Constant_range_defined_domain.....	105
10.4.9	Variable_range_defined_domain.....	105
10.4.10	Predicate_defined_domain.....	106
10.4.11	Functional_domain_restriction.....	107
10.4.12	Guarded_functional_domain.....	107
10.4.13	Simple_functional_domain.....	107

10.4.14	Library_expression_defined_value	107
10.4.15	Table_defined_value	108
10.4.16	Null_defined_value	109
10.5	ISO13584_domain_resource_schema function definitions	109
10.5.1	Collects_variables function	109
10.5.2	Collects_var_sem function	109
10.5.3	Used_tables_in_domain function	109
10.5.4	Used_variables_in_domain function	110
10.5.5	Variables_belong_to_assumes function	110
11	ISO13584_extended_dictionary_schema	110
11.1	Introduction to the ISO13584_extended_dictionary_schema	111
11.2	Fundamental concepts and assumptions for the ISO13584_extended_dictionary_schema	112
11.2.1	Dictionary structure	112
11.2.2	Class related elements	112
11.2.3	Supplier related elements	113
11.2.4	Three-fold description of dictionary elements	113
11.2.5	Unique identification of dictionary elements	113
11.2.6	Applicable elements	113
11.2.7	Visibility rule	114
11.2.8	Semantic relationships between classes	114
11.2.9	A priori semantic relationships and importation rule	115
11.2.10	Type checking for the tables referenced in the dictionary	115
11.3	ISO13584_extended_dictionary_schema constant definitions	116
11.3.1	Element_code_len	116
11.3.2	Dictionary_code_len	116
11.4	ISO13584_extended_dictionary_schema type definitions	116
11.4.1	Document_code_type	116
11.4.2	Program_library_code_type	116
11.4.3	Table_code_type	116
11.4.4	Absolute_URL_type	117
11.4.5	Dictionary_code_type	117
11.5	ISO13584_extended_dictionary_schema identification of a dictionary	117
11.6	ISO13584_extended_dictionary_schema overall architecture of a dictionary	118
11.7	Dictionary_in_standard_format	122
11.8	Data_exchange_specification_identification	123
11.9	Library_iim_identification	123
11.10	View_exchange_protocol_identification	124
11.11	ISO13584_extended_dictionary_schema entity definitions: additional entity instance types	124
11.11.1	Representation_type	124
11.11.2	Geometric_representation_context_type	125
11.11.3	Representation_reference_type	125
11.11.4	Program_reference_type	125
11.12	ISO13584_extended_dictionary_schema entity definitions: additional basic semantic units	126
11.12.1	Program_library_BSU	126
11.12.2	Table_BSU	126
11.12.3	Document_BSU	127
11.13	ISO13584_extended_dictionary_schema entity definitions: supplier BSU relationship	128
11.13.1	Supplier_program_library_relationship	128
11.14	ISO13584_extended_dictionary_schema entity definitions: class BSU relationships	128
11.14.1	Class_table_relationship	128
11.14.2	Class_document_relationship	129
11.15	ISO13584_extended_dictionary_schema entity definitions: properties of functional models and functional views	129
11.15.1	Representation_P_DET	129

11.16	ISO13584_extended_dictionary_schema entity definitions: specific dictionary elements.....	129
11.16.1	Supplier_related_dictionary_element.....	130
11.16.2	Class_related_dictionary_element.....	130
11.16.3	Program_library_element.....	130
11.17	ISO13584_extended_dictionary_schema entity definitions: class related elements.....	130
11.17.1	Table_element.....	131
11.17.2	RDB_table_element.....	131
11.17.3	Document_element.....	131
11.17.4	Document_element_with_http_access.....	132
11.17.5	Document_element_with_translated_http_access.....	132
11.17.6	Referenced_document.....	133
11.17.7	Referenced_graphics.....	133
11.18	ISO13584_extended_dictionary_schema entity definitions: feature class.....	134
11.19	ISO13584_extended_dictionary_schema entity definitions: a priori semantic relationship.....	134
11.20	ISO13584_extended_dictionary_schema entity definitions: functional model class.....	135
11.20.1	Abstract_functional_model_class.....	136
11.20.2	Functional_model_class.....	138
11.20.3	Fm_class_view_of.....	138
11.21	ISO13584_extended_dictionary_schema entity definitions: functional view class.....	139
11.21.1	Functional_view_class.....	140
11.21.2	Non_instantiable_functional_view_class.....	141
11.21.3	Specification of the range of a view control variable.....	141
11.22	ISO13584_extended_dictionary_schema entity definitions: item class a priori case of.....	142
11.22.1	Item_class_case_of.....	142
11.22.2	Component_class_case_of.....	142
11.22.3	Material_class_case_of.....	143
11.22.4	Feature_class_case_of.....	143
11.23	ISO13584_extended_dictionary_schema entity definitions: a posteriori semantic relationships.....	143
11.23.1	A_posteriori_semantic_relationship.....	143
11.23.2	A_posteriori_case_of.....	143
11.23.3	A_posteriori_view_of.....	144
11.24	ISO13584_extended_dictionary_schema entity definitions: table contents.....	145
11.24.1	Table_content.....	145
11.24.2	RDB_table_content.....	146
11.25	ISO13584_extended_dictionary_schema: RULE definitions.....	146
11.25.1	Representation_properties_for_model_and_view_rule rule.....	146
11.25.2	Allowed_named_type_usage_rule rule.....	147
11.25.3	Assert_oneof_for_table_rule rule.....	147
11.25.4	Assert_oneof_for_class_rule rule.....	147
11.25.5	No_forward_reference_from_table_rule rule.....	148
11.25.6	Imported_properties_are_visible_or_applicable_rule rule.....	148
11.25.7	Imported_data_types_are_visible_or_applicable_rule rule.....	148
11.25.8	Imported_tables_are_visible_or_applicable_rule rule.....	149
11.25.9	Imported_documents_are_visible_or_applicable_rule rule.....	149
11.26	ISO13584_extended_dictionary_schema: function definitions.....	149
11.26.1	Visible_properties function.....	150
11.26.2	Visible_types function.....	150
11.26.3	Visible_tables function.....	150
11.26.4	Visible_documents function.....	150
11.26.5	Applicable_properties function.....	151
11.26.6	Applicable_types function.....	151
11.26.7	Applicable_tables function.....	151
11.26.8	Retrieve_tables function.....	152
11.26.9	Applicable_documents function.....	152
11.26.10	Retrieve_documents function.....	152

11.26.11	Makes_reference_outside function.....	153
11.26.12	Prefix_ordered_class_list function.....	153
11.26.13	Functional_view_v_c_v function.....	155
11.26.14	Retrieve_functional_view_v_c_v function.....	155
11.26.15	Data_type_named_type function.....	155
11.26.16	Data_type_typeof function.....	155
11.26.17	Data_type_class_of function.....	156
11.26.18	Data_type_type_name function.....	156
11.26.19	Data_type_level_spec function.....	157
11.26.20	Data_type_level_value_typeof function.....	157
11.26.21	Simple_type_data_type function.....	157
11.26.22	Complex_type_data_type function.....	158
11.26.23	Compatible_subclass function.....	158
11.26.24	Compatible_types function.....	158
11.26.25	Ordered_index_value function.....	159
11.26.26	Makes_sub_list.....	159
11.26.27	Sub_list_until.....	160
11.26.28	Get_property_BSU_from_property_semantics function.....	160
11.26.29	Compatible_list_library_types_and_columns function.....	160
11.26.30	Data_type_non_quantitative_int_type function.....	161
11.26.31	Data_type_non_quantitative_code_type function.....	161
11.26.32	Applicable_properties_for_applicable_tables function.....	162
11.26.33	Superclass_of_item_is_item function.....	162
11.26.34	Compatible_content_and_specification function.....	162
11.26.35	Check_view_of_instance_datatype function.....	162
11.26.36	View_control_variables_attributes_belong_to_domain function.....	163
11.26.37	Created_view_is_functional_view function.....	163
11.26.38	Check_is_case_of_referenced_classes_definition function.....	163
12	ISO13584_library_content_schema.....	163
12.1	Introduction to the ISO13584_library_content_schema.....	164
12.2	Fundamental concepts and assumption for the ISO13584_library_content_schema.....	165
12.2.1	Class extension of non-leaf classes.....	165
12.2.2	Explicit description of class extensions.....	165
12.2.3	Implicit description of class extensions.....	166
12.2.4	Common pieces of information in implicit description and in explicit description of class extensions.....	166
12.2.5	Properties modeling in explicit description of class extensions.....	167
12.2.6	Typical usage of explicit description of class extensions.....	168
12.2.7	Properties modeling in implicit description of class extensions.....	170
12.2.8	Assemblies modeling in explicit description of class extensions.....	172
12.2.9	Assemblies modeling in implicit description of class extensions.....	173
12.2.10	Instances satisfying a class definition in an implicit description of a class extension.....	174
12.2.11	Mandatory support of the user selection process when implicit description of class extensions are used.....	176
12.3	ISO13584_library_content_schema constant definitions.....	179
12.3.1	Classification_value.....	179
12.4	ISO13584_library_content_schema: overall architecture of a library.....	179
12.5	Library_in_standard_format.....	180
12.6	Extension of a class.....	180
12.6.1	Class_extension.....	180
12.6.2	Opt_or_mand_property_BSU.....	181
12.6.3	Property_classification.....	181
12.6.4	Property_value_recommended_presentation.....	182
12.6.5	Model_class_extension.....	182
12.6.6	Explicit_model_class_extension.....	184
12.6.7	Explicit_item_class_extension.....	185
12.6.8	Explicit_functional_model_class_extension.....	185

12.6.9	Implicit_model_class_extension.....	189
12.6.10	Item_class_extension.....	191
12.6.11	Functional_model_class_extension.....	194
12.7	ISO13584_library_content_schema: RULE definitions.....	196
12.7.1	Assert_oneof_for_library_rule rule.....	196
12.7.2	Declared_created_views_are_created_rule rule.....	196
12.7.3	Complete_identification_for_instance_rule rule.....	197
12.7.4	Complete_identification_for_item_instance_rule rule.....	197
12.7.5	Complete_identification_for_model_instance_rule rule.....	198
12.7.6	All_views_available_for_each_component_rule rule.....	198
12.8	ISO13584_library_content_schema function definitions.....	199
12.8.1	Acyclic_class_extension_definition.....	199
12.8.2	Acyclic_order.....	199
12.8.3	Defined_domain function.....	199
12.8.4	Defined_derivation_function function.....	199
12.8.5	Allowed_properties function.....	200
12.8.6	Provided_properties_list function.....	200
12.8.7	Provided_properties_or_method_variables function.....	200
12.8.8	Selectable_properties_list function.....	200
12.8.9	Required_defined_properties function.....	200
12.8.10	Derived_properties_list function.....	201
12.8.11	Optional_properties_list function.....	201
12.8.12	Method_variables function.....	201
12.8.13	Gm_identification_characteristics_list function.....	201
12.8.14	Fm_free_model_properties_list function.....	202
12.8.15	Exists_super function.....	202
12.8.16	Super function.....	202
12.8.17	Is_in_v_c_v_range function.....	202
12.8.18	Get_v_c_v_range function.....	202
12.8.19	All_v_c_v_range_available function.....	203
12.8.20	Make_ordered_list_of_v_c_v_range function.....	203
12.8.21	Cdr_list function.....	203
12.8.22	Make_tuple function.....	203
12.8.23	Computable_set_of_created_views_from_model.....	204
12.8.24	Declared_created_views function.....	204
12.8.25	Created_views_by_methods function.....	204
12.8.26	In_typeof function.....	205
12.8.27	Referenced_veps_exist_in_supported_veps function.....	205
12.8.28	Referenced_protocols_exist_in_supported_protocols function.....	205
12.8.29	Required_properties_are_non_dependent_p_det function.....	205
12.8.30	Required_properties_are_imported_properties function.....	206
12.8.31	Same_order_for_properties function.....	206
12.8.32	All_properties_are_applicable function.....	206
12.8.33	Required_values_are_non_dependent_p_det function.....	206
12.8.34	Required_values_are_imported_properties function.....	207
12.8.35	Data_type_of_BSU function.....	207
12.8.36	Presentation_unit_is_correct function.....	207
12.8.37	Exists_representation_for_instanciable_view function.....	208
12.8.38	Is_provided_once_property_value function.....	208
12.8.39	Number_of_instance_representations.....	209
12.8.40	Correct_parameters_for_explicit_program function.....	209
12.8.41	Get_dic_item_instances_from_required_item_properties function.....	209
12.8.42	Get_list_of_required_properties function.....	210
12.8.43	Properties_projection_on_population function.....	210
12.8.44	All_views_available_for_components function.....	210
12.8.45	Available_components_views function.....	211
12.8.46	All_view_control_variables_belong_to_each_view function.....	211
12.8.47	Check_all_view_control_variables_belong_to_view function.....	211

	12.8.48 All_vcvs_belong_to_instance_identification function.....	212
	12.8.49 Same_string_values_translations_for_class_extension function.....	212
13	ISO13584_external_file_schema.....	212
13.1	Introduction to the ISO13584_external_file_schema.....	213
13.2	Fundamental concepts and assumptions for the ISO13584_external_file_schema.....	215
13.2.1	Representations of items.....	215
13.2.2	Explicit and implicit description of item representations.....	215
13.2.3	Support of user dialogue.....	216
13.2.4	Http files storage.....	216
13.2.5	Hyper-text link usage.....	217
13.2.6	Escape mechanism from document navigation to data retrieval and selection.....	217
13.2.7	Common Gateway Interface access.....	217
13.2.8	Common Gateway Interface implementation rule.....	219
13.3	ISO13584_external_file_schema constant definitions.....	219
13.3.1	Compiler_version_length.....	219
13.3.2	External_file_address_length.....	220
13.3.3	External_item_code_length.....	220
13.3.4	Http_file_name_length.....	220
13.3.5	Http_directory_name_length.....	220
13.4	ISO13584_external_file_schema type definitions.....	220
13.4.1	External_file_address.....	220
13.4.2	External_item_code_type.....	220
13.4.3	Http_file_name_type.....	221
13.4.4	Http_directory_name_type.....	221
13.4.5	MIME_type.....	221
13.4.6	MIME_subtype.....	222
13.4.7	IAB_RFC.....	222
13.4.8	Character_set_type.....	222
13.4.9	Content_encoding_type.....	222
13.4.10	Program_status.....	223
13.4.11	Program_reference_name_type.....	223
13.4.12	Compiler_version_type.....	223
13.4.13	Illustration_type.....	223
13.5	ISO13584_external_file_schema entity definitions: external_file_protocols.....	224
13.5.1	External_file_protocol.....	224
13.5.2	Standard_protocol.....	225
13.5.3	Non_standard_protocol.....	225
13.5.4	Data_protocol.....	225
13.5.5	Program_protocol.....	226
13.5.6	Simple_program_protocol.....	226
13.5.7	Standard_simple_program_protocol.....	226
13.5.8	Non_standard_simple_program_protocol.....	227
13.5.9	Linked_interface_program_protocol.....	227
13.5.10	Standard_data_protocol.....	227
13.5.11	Non_standard_data_protocol.....	228
13.5.12	Http_protocol.....	228
13.6	ISO13584_external_file_schema entity definitions: dictionary external items.....	228
13.6.1	External_item.....	228
13.6.2	Dictionary_external_item.....	229
13.6.3	Supplier_BSU_related_content.....	229
13.6.4	Program_library_content.....	229
13.6.5	Class_BSU_related_content.....	230
13.6.6	Document_content.....	230
13.7	ISO13584_external_file_schema entity definition: class extension external items.....	230
13.7.1	Class_extension_external_item.....	231
13.7.2	Representation_reference.....	231
13.7.3	Program_reference.....	231

13.7.4	Dialogue_resource	232
13.7.5	Message	233
13.7.6	Illustration	233
13.7.7	A6_illustration	233
13.7.8	A9_illustration	234
13.8	ISO13584_external_file_schema entity definition: property_value_external_item	234
13.9	ISO13584_external_file_schema rule definition	234
13.9.1	Unique_http_file_name_per_supplier_element_rule rule	234
13.9.2	Unique_http_directory_name_per_supplier_rule rule	235
13.9.3	No_http_directory_for_supplier_related_file_rule rule	235
13.9.4	Http_directory_refers_to_bsu_related_class_rule rule	235
13.9.5	Http_directory_refers_to_class_extension_rule rule	236
13.9.6	Illustration_is_not_a_referenced_graphics_rule rule	236
13.10	ISO13584_external_file_schema entity definitions: external content	236
13.10.1	External_content	237
13.10.2	Translated_external_content	237
13.10.3	Not_translated_external_content	238
13.10.4	Not_translatable_external_content	238
13.10.5	Language_specific_content	238
13.10.6	External_file_unit	239
13.10.7	Http_file	239
13.10.8	Http_class_directory	240
13.11	ISO13584_external_file_schema function definitions	241
13.11.1	Supplier_associated_http_files	241
13.11.2	Control_compiler_version_format	241
14	ISO13584_method_schema	241
14.1	Introduction to the ISO13584_method_schema	242
14.2	Fundamental concepts and assumptions for the ISO13584_method_schema	243
14.3	ISO13584_method_schema type definitions	244
14.3.1	Accessible_variable_for_method	244
14.3.2	Assignment_allowed_variable	245
14.3.3	Control_allowed_variable	246
14.4	ISO13584_method_schema entity definitions	246
14.4.1	Method	246
14.4.2	Method_specif	247
14.4.3	Method_body	248
14.4.4	Method_statement	249
14.4.5	Guarded_statement	249
14.4.6	Simple_statement	250
14.4.7	Null_statement	250
14.4.8	Modelling statement	250
14.4.9	Set_reference_lcs	250
14.4.10	Begin_set	251
14.4.11	Close_set	252
14.4.12	Set_2d_relative_view_level	252
14.4.13	Predefined_representation_call_statement	253
14.4.14	Send_representation_statement	253
14.4.15	Send_representation_reference_statement	255
14.4.16	Call_program_statement	256
14.4.17	Assignment_statement	258
14.4.18	Sub_object_view_statement	259
14.4.19	Referenced_sub_item_view_statement	260
14.4.20	Constructed_sub_model_view_statement	261
14.5	ISO13584_method_schema rules definitions	262
14.5.1	Created_view_v_c_v_rule rule	262
14.5.2	V_c_v_values_set_and_created_view_v_c_v_set_equality_rule rule	262
14.5.3	No_v_c_v_in_assigned_variables_set_rule rule	262
14.6	ISO13584_method_schema function definitions	262

14.6.1	Checks_classes_in_path function	263
14.6.2	Checks_applicable_properties_in_path function	263
14.6.3	same_view_model_method	263
14.6.4	self_property_value_semantics_is_item_class	264
15	Conformance requirements	264
16	Exchange of general model classes: library integrated information model 24-1	266
16.1	ISO13584_g_m_iim_schema short listing	267
16.2	ISO13584_g_m_iim_schema global rule definitions	269
16.2.1	At_most_one_dictionary_rule rule	269
16.2.2	Class_associated_items_rule rule	269
16.3	Conformance class requirements	270
16.3.1	Conformance class 0	270
16.3.2	Conformance class 1	270
16.3.3	Conformance class 1E	271
16.3.4	Conformance class 2	271
16.3.5	Conformance class 2E	271
16.3.6	Conformance class 3	271
16.3.7	Conformance class 3E	272
16.3.8	Conformance class 4	272
16.3.9	Conformance class 4E	272
16.3.10	Conformance class 5	272
16.3.11	Conformance class 5E	273
16.3.12	Conformance class 6	273
16.3.13	Conformance class 6E	273
17	Exchange of functional model classes: library integrated information model 24-2	274
17.1	ISO13584_f_m_iim_schema short listing	275
17.2	ISO13584_f_m_iim_schema global rule definitions	277
17.2.1	Exactly_one_dictionary_rule rule	277
17.2.2	Class_associated_items_rule rule	277
17.2.3	Supplier_associated_items_rule rule	278
17.3	Conformance class requirements	278
17.3.1	Conformance class 1	278
17.3.2	Conformance class 1E	278
17.3.3	Conformance class 2	279
17.3.4	Conformance class 2E	279
17.3.5	Conformance class 3	279
17.3.6	Conformance class 3E	280
17.3.7	Conformance class 4	280
17.3.8	Conformance class 4E	280
17.3.9	Conformance class 5	280
17.3.10	Conformance class 5E	281
17.3.11	Conformance class 6	281
17.3.12	Conformance class 6E	281
18	Exchange of functional view classes: library integrated information model 24-3	282
18.1	ISO13584_f_v_iim_schema short listing	283
18.2	ISO13584_f_v_iim_schema global rule definitions	283
18.2.1	Exactly_one_dictionary_rule rule	283
18.2.2	Class_associated_items_rule rule	284
18.3	Conformance class requirements	284
18.3.1	Conformance class 1	284
18.3.2	Conformance class 1E	284
18.3.3	Conformance class 2	285
18.3.4	Conformance class 2E	285
	Bibliography	286
	Annex A (normative) Short names of entities defined in this part	287

Annex B (normative) Information object registration	293
Annex C (normative) ISO13584_g_m_iim_library_implicit_schema expanded listing.....	295
Annex D (normative) ISO13584_g_m_iim_schema short names of entities.....	297
Annex E (normative) Standard data requirements for the library integrated information model 24-1.....	298
Annex F (normative) Implementation method specific requirements for the library integrated information model 24-1.....	304
Annex G (normative) ISO13584_f_m_iim_library_implicit_schema expanded listing.....	305
AnnexH (informative) ISO13584_f_m_iim_schema short names of entities	307
AnnexI (normative) Standard data requirements for the library integrated information model 24-2.....	308
AnnexJ (normative) Implementation method specific requirements for the library integrated information model 24-2.....	314
AnnexK (normative) ISO13584_f_v_iim_library_implicit_schema expanded listing.....	315
AnnexL (informative) ISO13584_f_v_iim_schema short names of entities.....	317
AnnexM (normative) Standard data requirements for the library integrated information model 24-3.....	318
AnnexO (informative) Logical description of the compiling process of ISO13584- conformant dictionaries and libraries	324
AnnexP (informative) Commented example of Parts Library physical files.....	327
AnnexQ (informative) Guidelines for creating functional model classes.....	354
AnnexR (informative) EXPRESS-G diagrams.....	356
AnnexS (informative) Notational Conventions and Generic Grammar for URL-encoded strings ..	372

Foreword

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

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

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

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

[ISO 13584-24](#) was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

[ISO 13584](#) consists of the following parts, under the general title *Industrial automation systems and integration — Parts library*:

- *Part 1: Overview and fundamental principles*
- *Part 20: Logical resource: Logical model of expressions*
- *Part 24: Logical resource: Logical model of supplier library*
- *Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content*
- *Part 26: Logical resource: Information supplier identification*
- *Part 31: Implementation resources: Geometric programming interface*
- *Part 42: Description methodology: Methodology for structuring part families*
- *Part 101: Geometrical view exchange protocol by parametric program*
- *Part 102: View exchange protocol by [ISO 10303](#) conforming specification*

The structure of this International Standard is described in ISO 13584-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 10 to 19 specify the conceptual descriptions;
- Parts 20 to 29 specify the logical resources;
- Parts 30 to 39 specify the implementation resources;
- Parts 40 to 49 specify the description methodology;
- Parts 100 to 199 specify the view exchange protocol.

Should further parts of [ISO 13584](#) be published, they will follow the same numbering pattern.

Introduction

[ISO 13584](#) is an International Standard for the computer-interpretable representation and exchange of parts 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.

This International Standard is organized 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 and view exchange protocol. The series are described in ISO 13584-1. This part of [ISO 13584](#) is a member of the logical resources series.

This part of [ISO 13584](#) specifies the generic resources needed for supplier library modelling and exchange. It also provides the EXPRESS integrated information models that permit the exchange of libraries that consist either of definitions of families of parts, representations of families of parts, or definitions of new representation categories that may be provided for any family of parts. Knowledge of EXPRESS as defined in [ISO 10303-11](#) is required to understand this part of [ISO 13584](#). Basic knowledge of [ISO 13584-20](#) and [ISO 13584-42](#) is also required.

Industrial automation systems and integration — Parts library —

Part 24: Logical resource: Logical model of supplier library

1 Scope

This part of [ISO 13584](#) specifies generic EXPRESS resource constructs that support the description of different kinds of information about supplier libraries. It also contains a set of integrated EXPRESS information models for representing supplier libraries for the purpose of exchange. These integrated information models integrate EXPRESS resource constructs from different parts of [ISO 13584](#) and [ISO 10303](#) into a single schema. Supplier libraries may consist of definitions or representations of families of parts. Supplier libraries may also define new representation categories. Supplier libraries may consist only of dictionary elements, or they may also contain specifications of permitted instances.

When used together with view exchange protocols, these integrated information models also permit the exchange of one or several representation categories for the parts defined in a parts library.

NOTE 1 View exchange protocols are defined in the view exchange protocol series of [ISO 13584](#).

The following are within the scope of this part of [ISO 13584](#):

- Generic resource constructs for representing hierarchies of families of parts. The parts in the families may be components or assembled parts, and may be abstract parts or physical parts.
- Generic resource constructs for representing implicitly the definitions of the different parts that belong to a family of parts.
- Generic resource constructs for representing the different kinds of possible representations of the different parts that belong to a family of parts.
- Generic resource constructs for representing families of materials, together with their definitions and possible representations.
- Library integrated information models that gather generic resource constructs from different parts of [ISO 13584](#) and [ISO 10303](#) into one single schema for representing supplier libraries for the purpose of exchange. The supplier libraries may consist either of definitions of families of parts, or of representations of families of parts or of definitions of new representation categories that may be provided for any family of parts.

The following are outside the scope of this part of [ISO 13584](#):

- Description of assembled parts that may contain an unlimited number of constituent components.
- Specification of a software system able to manage supplier libraries represented according to the information models defined in this part of [ISO 13584](#).
- Description of the different representation categories that a supplier library may contain.

NOTE 2 The description of the different representation categories that a supplier library may contain are defined in the view exchange protocol series of parts of [ISO 13584](#).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[ISO 3166:1993](#), *Codes for the representation of names of countries*

[ISO 6093:1985](#), *Information processing — Representation of numerical values in character strings for information interchange*

ISO/IEC 8824-1:2003, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

[ISO 8859-1:1987](#), *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/IEC 9075:1992](#), *Information technology — Database languages — SQL*

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

ISO 10303-21:2002, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*

ISO 10303-41:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*

ISO 10303-42:2003, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

ISO 10303-43:2003, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures*

ISO 10303-227:2001, *Industrial automation systems and integration — Product data representation and exchange — Application protocol: Plant spatial configuration*

ISO 13584-1:2001, *Industrial automation systems and integration — Parts library — Part 1: Overview and fundamental principles*

[ISO 13584-20:1998](#), *Industrial automation systems and integration — Parts library — Part 20: Logical resource: Logical model of expressions*

ISO 13584-26:2000, *Industrial automation systems and integration — Parts library — Part 26: Logical resource: Information supplier identification*

ISO 13584-31:1999, *Industrial automation systems and integration — Parts library — Part 31: Implementation resources: Geometric programming interface*

[ISO 13584-42:1998](#), *Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families*

[IEC 61360-2:2002](#), *Standard data element types with associated classification scheme for electric components — Part 2: EXPRESS dictionary schema*

IAB RFC 1700:1994, *Internet architecture board internet standard (STD 2): Assigned numbers*

IAB RFC 1739:1994, *Internet architecture board proposed standard protocol: A primer on Internet and TCP/IP tools*

IAB RFC 1808:1995, *Internet architecture board proposed standard protocol: Relative uniform resource locators (URL)*

IAB RFC 1866:1995, *Internet architecture board proposed standard protocol: Hypertext markup language 2.0 (HTML-2.0)*

IAB RFC 2045:1996, *Internet architecture board draft standard protocol: Multipurpose internet mail extensions (MIME)*

IAB RFC 2049:1996, *Internet architecture board draft standard protocol: MIME conformance criteria (MIME-CONF)*

IAB RFC 2068:1997, *Internet architecture board proposed standard protocol: Hypertext transfer protocol HTTP/1.1 (HTTP-1.1)*

IAB RFC 2231:1997, *Internet architecture board Proposed standard protocol: MIME parameter value and encoded word extensions: Character sets, languages and continuations*

IAB RFC 2400:1998, *Internet architecture board internet standard (STD 1): Internet official protocol standard*

3 Terms, definitions and abbreviations

For the purposes of this document, the following terms and definitions apply. Some of these terms and definitions are repeated for convenience from:

- [ISO 10303-1:1994](#);
- [ISO 10303-11:1994](#);
- ISO 13584-1:2001;
- ISO 13584-31:1999.
- [ISO 13584-42:1998](#).

3.1

absolute uniform resource locator

a string, the content of which uniquely identifies a network resource over the Internet

Note 1 to entry: The structure of absolute uniform resource locators are defined in IAB RFC 1738 [\[2\]](#).

Note 2 to entry: The information represented in an absolute uniform resource locator includes the following:

- the protocol to be used by the Internet client to access the resource, and
- the Internet address of the Internet server.

3.2

abstract part

a part that is only defined by a partial specification and that cannot be materially provided by the organisation that defines the specification, e.g., International Standard, functional specification (compare with a physical part)

[SOURCE: ISO 13584-1]

3.3

applicable property

a property that is defined for some family of parts and that shall apply to any part that belongs to this family of parts

EXAMPLE For a generic family of screws, the threaded diameter is an applicable property. This characteristic applies to any screw.

3.4
application

a group of one or more processes creating or using product data

[SOURCE: ISO 10303-1, definition 3.2.2]

3.5
application context

the conditions that define the intended use of product data within an application

[SOURCE: ISO 10303-1, definition 3.2.4]

3.6
application programming interface

API

a set of functions that may be triggered by a program

3.7
application protocol

AP

a part of [ISO 10303](#) that describes the use of ([ISO 10303](#)) integrated resources satisfying the scope and information requirements for a specific application context

[SOURCE: ISO 10303-1, definition 3.2.7]

3.8
assembled item

an item that is defined as a composition of other items

3.9
atomic item

an item that is not defined as a composition of other items

Note 1 to entry: A part that consists of several subassemblies may be described as an atomic item if its class definition does not define its constituent subassemblies.

3.10
augmented Backus-Naur form

ABNF

the augmented version of Backus-Naur syntax notation defined in clause 2 of IAB RFC 1808

Note 1 to entry: The notational conventions used in ABNF are summarised for convenience in the informative annex T.

3.11
basic semantic unit

BSU

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

[SOURCE: ISO 13584-42, definition 3.4.1]

3.12
characteristic of a part

part characteristic

a constant property, characteristic of a part, of which the value is fixed once the part is defined

Note 1 to entry: Changing the value of a characteristic of a part would mean changing the part.

EXAMPLE For a ball-bearing, the inner and outer diameters are part characteristics.

3.13

class extension

the set of all instances satisfying the class definition

3.14

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

[SOURCE: ISO 13584-42, definition 3.4.2]

Note 1 to entry: Class valued properties may be used to capture some commonality between different families when such a commonality is not captured by the hierarchy structure.

3.15

common dictionary schema

the information model for a dictionary, using the EXPRESS modelling language, resulting from a joint effort between ISO TC184/SC4/WG2 and IEC SC3D

[SOURCE: ISO 13584-42, definition 3.4.3]

Note 1 to entry: The common dictionary schema is specified in [IEC 61360-2](#). Its content is provided in the informative annex D of [ISO 13584-42](#).

3.16

completely defined instance

any particular instance of a class extension (compare with: partially defined instance)

3.17

conformance class

a subset of a standard for which conformance may be claimed

Note 1 to entry: Adapted from [ISO 10303-1](#).

3.18

conformance requirement

a precise text definition of a characteristic required to be present in a conforming implementation

[SOURCE: ISO 10303-1, definition 2.1.14]

3.19

conforming implementation

an implementation which satisfies the conformance requirements defined by one or several conformance classes of a standard

3.20

conformity

conformance

the fulfilment by an implementation of all specified requirements

[SOURCE: ISO 10303-31, definition 3.2.25]

3.21

context

the circumstances relevant to something under consideration

3.22

context-dependent characteristic of a part

a property of a part whose value depends on some context parameter(s)

Note 1 to entry: For a given part a context-dependent characteristic is mathematically defined as a function whose domain is defined by some context parameter(s) that defines the part environment.

EXAMPLE For a ball-bearing, the life-time is a context-dependent characteristic that depends on the radial load, the axial load and the rotational speed.

3.23

context parameter

a variable of which the value characterises the context in which it is intended to insert a part

EXAMPLE The dynamic-load applied to a bearing is a context parameter for this bearing.

3.24

data

a representation of facts, concepts or instructions in a formal manner suitable for communication, interpretation, or processing by human beings or computers

[SOURCE: ISO 10303-1, definition 3.2.14]

3.25

data element type

DET

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

[SOURCE: ISO 13584-42, definition 3.4.4]

3.26

data exchange

the storing, accessing, transferring, and archiving of data

[SOURCE: ISO 10303-1, definition 3.2.15]

3.27

data type

a domain of values

[SOURCE: ISO 10303-11, definition 3.2.4]

3.28

database oriented navigation

within a user library, a navigation that results from querying the data delivered by library data suppliers as library delivery files

3.29

definition table

the definition of the set of parts that belong to a family of parts by means of a two-dimensional array finite or infinite where each row defines a part and where each column describes the corresponding values of a part characteristics

Note 1 to entry: In [ISO 13584](#), only those family of parts that are associated with a class extension include the description of their definition tables.

Note 2 to entry: In [ISO 13584](#), a definition table may be defined either explicitly or implicitly by means of basic domains, set operators and relational algebra operators.

3.30

derivation function

relationship serving to compute the value of a property from the values of other properties

3.31

derived characteristics

in a family of parts associated with a definition table, a set of part characteristics that does not belong to the (library data supplier defined) key of this definition table

Note 1 to entry: A functional dependency exists between the identification characteristics and the derived characteristics. In [ISO 13584](#), this functional dependency is represented by a derivation function.

3.32

dictionary

a table consisting of a series of entries. One meaning corresponds to each entry in the dictionary and one dictionary entry identifies one single meaning

[SOURCE: ISO 13584-1]

Note 1 to entry: In [ISO 13584](#), the kinds of meaning intended to constitute dictionary entries are: supplier, class, property, program library, type, table and document.

Note 2 to entry: In [ISO 13584](#), the information that represents a dictionary entry is split into three entities: a *basic_semantic_unit* (BSU), that provides for reference, a **dictionary_element** that describes the dictionary entry by means of attributes, and, possibly, a **content_item** entity that describes the dictionary entry by describing its content.

3.33

dictionary data

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

[SOURCE: ISO 13584-42, definition 3.4.6]

Note 1 to entry: The dictionary data should be exchanged using the common dictionary schema.

3.34

dictionary element

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

[SOURCE: ISO 13584-42, definition 3.4.7]

3.35

document oriented navigation

within a user library navigation that follows hypertext links to navigate between documents delivered by library data suppliers as library external files

3.36

entity

a class of information defined by common properties

[SOURCE: ISO 10303-11, definition 3.2.5]

3.37

entity data type

a representation of an entity. An entity data type establishes a domain of values defined by common attributes and constraints

[SOURCE: ISO 10303-11, definition 3.2.6]

3.38

entity (data type) instance

a named unit of data that represents a unit of information within the class defined by an entity. It is a member of the domain established by an entity data type

[SOURCE: ISO 10303-11, definition 3.2.7]

3.39

exchange structure

a computer-interpretable format used for storing, accessing, transferring, and archiving data

[SOURCE: ISO 10303-1, definition 3.2.17]

3.40

family of parts

a simple or generic family of parts

3.41

feature

an aspect of an item that can be captured by a class structure and set of properties and that cannot exist independently of the item

EXAMPLE 1 A form feature is an aspect of a part that conforms to some preconceived shape stereotype associated to dimensioning properties. It may be represented as an instance of a class that captures this shape stereotype.

EXAMPLE 2 In a piping component, an outlet is an aspect of a part that conforms to some preconceived function stereotype that is associated with properties (e.g., its name, its role). It may be represented as a feature.

3.42

functional model of a part

the library data that represent one representation category of a part in an integrated library

[SOURCE: ISO 13584-1]

3.43

functional view of a part

a data that represents one representation category of a part in product data

[SOURCE: ISO 13584-1]

Note 1 to entry: The structure of a functional view does not depend on the part it represents.

3.44

generic family of parts

a grouping of simple or generic families of parts done for purposes of classification or for factoring common information

3.45

hypertext markup language

HTML

a particular implementation of [ISO 8879](#) (SGML) that enables access to the network resources available on the Internet

Note 1 to entry: The current version of HTML, HTML/2.0, is defined by IAB RFC 1866.

Note 2 to entry: An HTML document encapsulates uniform resource locators (URL) as a means of accessing network resources available on the Internet.

3.46

general model of a part

the library data that carries the definition and identity of a part in an integrated library

[SOURCE: ISO 13584-1]

3.47

http file

a MIME-like file that may contain reference to other Internet resources by means of http URLs

Note 1 to entry: http URLs are URLs used to locate network resources via the HTTP protocol. Their particular syntax and semantics are defined in the RFC that specifies the HTTP protocol.

Note 2 to entry: The current version of the HTTP protocol, HTTP/1.1, is defined by IAB RFC 2068.

3.48
hypertext transfer protocol
HTTP

a particular application-level network protocol between an Internet server and an Internet client defined by a RFC from the IAB

Note 1 to entry: The current version of the HTTP protocol, HTTP/1.1, is defined by IAB RFC 2068.

3.49
identification characteristics

a simple family of parts associated with a definition table, a set of part characteristics that constitute the (library data supplier-defined) key of this definition table

Note 1 to entry: When there exist several sets of part characteristics that might constitute a key of the definition table (candidate keys), the particular set that defines the identification characteristics is chosen by the library data supplier.

Note 2 to entry: Identification characteristics identify a part within its family.

Note 3 to entry: It is not allowed by [ISO 13584](#) to reuse the same values of the identification characteristics at any time for two different parts, i.e., for two parts of which some non-identification characteristics are different. If such a situation is anticipated, some additional identification characteristics, such that a version, shall be added to discriminate both parts.

Note 4 to entry: A functional dependency exists between the identification characteristics and the other part characteristics. In [ISO 13584](#), this functional dependency is represented by a derivation function.

3.50
implementation method

a technique used by computers to exchange data that is described using the EXPRESS data specification language

Note 1 to entry: Adapted from [ISO 10303-1](#).

3.51
implementation resources

the capabilities of a software system that shall be available to claim conformance to a particular conformance class of a view exchange protocol or both view exchange protocol and library integrated information model

3.52
information

facts, concepts or instructions

[SOURCE: ISO 10303-1, definition 3.2.20]

3.53
information model

a formal model of a set of facts, concepts or instructions to meet a specific requirement

[SOURCE: ISO 10303-1, definition 3.2.21]

3.54
instance

a named value

[SOURCE: ISO 10303-11, definition 3.2.8]

3.55
integrated library

operational system consisting of a library management system and a user library

[SOURCE: ISO 13584-1]

3.56

Internet

the network that complies with the standard rules defined by the Internet Architecture Board (IAB)

3.57

Internet architecture board

IAB

the organisation that specifies by means of Request For Comments (RFC) the standard rules that shall be followed by the Internet client and Internet server

Note 1 to entry: Current versions of standard documents from the Internet Architecture Board (IAB) may be found on the Internet Official Server at the following uniform resource locator: <http://DS.INTERNIC.NET>.

3.58

Internet assigned numbers authority

IANA

the central coordinator for the assignment of unique parameter values for Internet protocols

Note 1 to entry: Currently assigned values for the various series of protocol parameters for the Internet protocol suite are currently available as: <ftp://ftp.isi.edu/in-notes/iana/assignments/>.

3.59

Internet client

program that establishes connections over the Internet for the purpose of sending requests

Note 1 to entry: Requests conformant with the hypertext transfer protocol are defined by IAB RFC 2068.

3.60

Internet server

program that accepts connections in order to service requests by sending back responses

Note 1 to entry: Requests conformant with the hypertext transfer protocol are defined by IAB RFC 2068.

3.61

is-a relationship

the inheritance relationship defined in the object oriented paradigm

Note 1 to entry: In [ISO 13584](#) the is-a relationship holds between a family of parts and a generic family of parts to which the former family belongs.

3.62

is-case-of relationship

a relationship providing a formal expression of the fact that an object conforms to the partial specification defined by another object

Note 1 to entry: In [ISO 13584](#), all the properties and data types visible or applicable for some family of parts may be imported by all the families of parts that declare to be case-of the former family. These properties and data types may then be used to describe the latter families.

3.63

is-part-of

the aggregation part/whole relationship

Note 1 to entry: In [ISO 13584](#) the is-part-of relationship holds between a family of constituent parts and a family of assembled parts to which the constituent parts belongs.

3.64

is-view-of

a relationship providing a formal expression of the fact that an object is a representation of another object according to a particular perspective

EXAMPLE A set of geometric entities might provide a drafting representation of a particular screw. If both the set of geometric entities and the particular screw are represented as object, the is-view-of relationship holds between the former object and the latter object (in a drafting perspective).

3.65

item

a thing that can be captured by a class structure and a set of properties

[SOURCE: ISO 13584-42, definition 3.4.8]

3.66

library

See: Parts *library*, *supplier library*, *user library*

[SOURCE: ISO 13584-1]

3.67

library data supplier

an organisation that delivers a supplier library in the standard format defined in [ISO 13584](#) and is responsible for its content

[SOURCE: ISO 13584-1]

3.68

library delivery file

a population of EXPRESS entity instances conforming to a library integrated information model and represented according to one of the implementation methods specified in [ISO 10303](#)

Note 1 to entry: A library delivery file specifies the structure and the content of a supplier library. It may reference library external files.

3.69

library end-user

the user of an integrated library

[SOURCE: ISO 13584-1]

Note 1 to entry: The library end-user:

- consults the data contained in the library;
- selects a given part;
- requests the transmission of a selected view of this part from the library system.

3.70

library exchange context

the set of one library delivery file and zero, one or more library external files that represent together a supplier library

3.71

library external file

a file, referenced from a library delivery file, that contributes to the definition of a supplier library

Note 1 to entry: The structure and the format of a library external file is specified in the library delivery file that references it.

3.72

library integrated information model

LIIM

an EXPRESS schema that integrates resource constructs from different EXPRESS schemas for representing supplier libraries for the purpose of exchange and that is associated with conformance requirements

Note 1 to entry: Three library integrated information models are defined in this part of [ISO 13584](#) for representing different kinds of supplier libraries.

3.73

library management system

LMS

a software system enabling the library end-user to use the content of an integrated library

[SOURCE: ISO 13584-1]

Note 1 to entry: This software system is not standardised.

3.74

library part

a part associated with a set of data that represents it in a library

[SOURCE: ISO 13584-1]

3.75

library part data

the data that represent a part in a library

[SOURCE: ISO 13584-1]

3.76

library specification of a class

the explicit representation of a class extension in a supplier library

Note 1 to entry: In the [ISO 13584](#) series, every class is intentionally defined through a dictionary element. A dictionary element allows defining a kind of part, characterized by abstract properties. A class extension enables the library data supplier to define those parts of the kind defined by a dictionary element that are existing in some context. They may be e.g., sold by the library data supplier.

3.77

local coordinate system

LCS

an orthogonal right-handed coordinate system used to orientate and to locate geometrical entities in space

[SOURCE: ISO 13584-31]

3.78

mandatory property

a property that shall have a value in any completely defined instance of a class (compare with: optional property)

Note 1 to entry: As in the EXPRESS language, a NULL value for a property in an instance is represented by the fact that this property is associated with no value. No difference is made, at the data model level, between a still non-assigned value, and a NULL value.

3.79

MIME-like file

a file whose format is described by the MIME-defined set of fields of information

Note 1 to entry: Standard values for MIME-defined fields of information may be registered by IANA. In the context of [ISO 13584-24](#), only such registered values are allowed for use to describe the format of files exchanged as library external file.

Note 2 to entry: IANA registration procedures for MIME-related facilities are defined by RFCs from the IAB. They are currently defined by IAB RFC 2048 [\[5\]](#).

Note 3 to entry: Currently assigned values for the various series of protocol parameters for the Internet protocol suite are currently available as: <ftp://ftp.isi.edu/in-notes/iana/assignments/>.

3.80

multi-purpose Internet mail extensions

MIME

a set of fields of information that enables one to specify the format of a file on the Internet

Note 1 to entry: The MIME set of fields of information, and their meaning, are currently defined by IAB RFC 2045, IAB RFC 2046 [\[3\]](#), IAB RFC 2047 [\[4\]](#), IAB RFC 2049, and IAB RFC 2231.

Note 2 to entry: Particular values for MIME fields of information may be registered by IANA. Such values are intended to be recognised by any Internet agent.

3.81

network resource

an addressable unit of information or service that can be accessed through a network

3.82

object view coordinate

OVC

a Cartesian coordinate system used to describe geometric representations of library parts

3.83

optional property

a property that does not need a value for some completely defined instance of some class

EXAMPLE Consider the class that describes an assembled part consisting of bolt + nut + optional washer components. It contains a characteristic, called *the_washer*, whose type is defined by a washer class. For those instances of this class that do not contain a washer, the *the_washer* characteristic is not assigned a value.

3.84

part

a material or functional element that is intended to constitute a component of different products

[SOURCE: ISO 13584-1]

3.85

partially defined instance

an abstraction of several different instances of a class extension that may prove useful to capture in some state of a design process

Note 1 to entry: This part of [ISO 13584](#) does not specify whether partially defined instances may be created by a library in a user modelling system. This is implementation dependent.

3.86

parts library

an identified set of data and possibly programs which may generate information about a set of parts

[SOURCE: ISO 13584-1]

3.87

physical part

a part that can exist in several equivalent copies and which is capable of being supplied by the library data supplier who describes the library data for this part (compare to: abstract part)

[SOURCE: ISO 13584-1]

3.88

population

a collection of entity data type instances

[SOURCE: ISO 10303-11, definition 3.2.11]

3.89

product

a thing or substance produced by a natural or artificial process

[SOURCE: ISO 10303-1, definition 3.2.26]

3.90

product data

a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers

[SOURCE: ISO 10303-1, definition 3.2.27]

3.91

property

an information that may be represented by a data element type

[SOURCE: ISO 13584-42, definition 3.4.10]

3.92

relative uniform resource locator

a simplified URL that uniquely identifies a network resource provided by the same Internet server as the one which served the document where the URL is used

Note 1 to entry: A relative uniform resource locator does not contain the location of the Internet server.

Note 2 to entry: The relative URL content and structure is currently defined by IAB RFC 1808.

3.93

representation

a description, drawing or depiction of something

[SOURCE: ISO 10303-227]

3.94

representation category

a concept used to distinguish between different possible user requirements regarding a part representation

[SOURCE: ISO 13584-1]

Note 1 to entry: In the model defined in this International Standard, this distinction is formally expressed in terms of a view logical name and the view control variables.

3.95

representation property

a property of a representation of a part that is not a part characteristics

EXAMPLE In a functional model class that represents some simulation model for the different parts of a parts family, all the coefficients of these simulation models are representation properties. Unlike part characteristics, the values of these properties may be changed for a given part without changing the part. For instance, It is the case when the coefficients are more accurately computed.

3.96

request for comments

RFC

a document issued by the Internet Architecture Board(IAB) to specify a rule to be followed on the Internet

Note 1 to entry: This part of [ISO 13584](#) references, in its Normative reference clauses, several IAB RFCs for representing and accessing documents which constitute provisions of this part of [ISO 13584](#). Parties to agreements based on this part of [ISO 13584](#) are encouraged to investigate the possibility of applying the most recent RFCs released by the IAB for representing documents associated with a supplier library.

3.97

resource construct

the collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of data

Note 1 to entry: Adapted from [ISO 10303-1](#).

3.98

simple family of parts

a set of parts of which each part may be described by the same group of properties

3.99

standard data

a requirement on a software system defined by means of EXPRESS entity (data type) instances that are supposed to be recognised by this software system

3.100

standardised identification hierarchy

a dictionary data that is defined by a standardisation organisation

[SOURCE: ISO 13584-42, definition 3.4.11]

3.101

structured query language

SQL

the query language for relational database defined by [ISO/IEC 9075](#)

3.102

supplier library

a set of data, and possibly of programs, for which the supplier is defined and that describes in the standard format defined in [ISO 13584](#) a set of parts and/or a set of representation of parts

[SOURCE: ISO 13584-1]

3.103

uniform resource locator

URL

a mechanism specified by the Internet Architecture Board(IAB) to uniquely identify a network resource on the Internet

Note 1 to entry: An URL may be either an absolute URL or a relative URL.

Note 2 to entry: The URL mechanism is currently defined by IAB RFC 1738 [\[2\]](#) and IAB RFC 1808.

3.104

user library

the information that results from the integration of one or more supplier libraries by the library management system and possibly from a later adaptation performed by the user

[SOURCE: ISO 13584-1]

3.105

user modeling system

a software system enabling the library end-user to use the part representations generated by an integrated library

3.106

view control variable

a variable of enumerated type, that may be associated with a view logical name and intended to further specify the perspective adopted by the user regarding a part. (e.g., for geometry: 2D, wire frame, solid)

[SOURCE: ISO 13584-1]

3.107

view exchange protocol

VEP

a part of [ISO 13584](#) that describes the use of resource constructs and of representation transmission interfaces that satisfy the information requirement for the exchange of one representation category of parts

3.108

view logical name

the identifier of a representation category corresponding to a perspective that can be adopted by a user regarding a part (e.g., geometry, inertia, kinematics, etc.)

[SOURCE: ISO 13584-1]

3.109

visible property

a property that is defined for some family of parts and that may or not apply to the different parts of this family of parts

EXAMPLE For a generic family of screws, the non-threaded length is a visible property: it is clearly defined for any screw, but only those screws with a non-threaded part have a value for this property.

4 Structure of ISO 13584-24

[ISO 13584-24](#) has three main parts.

- The generic resources part provides resource constructs that are generic in nature. They are intended to be used both inside and outside the [ISO 13584](#) Standard series. This intent was taken into account in their design.
- The parts library specific resources part provides resource constructs that are specific to the parts library application domain. They were not designed with the intent to be used outside the [ISO 13584](#) Standard series.
- The library integrated information models part provides EXPRESS schemas that integrate resource constructs from the previous schemas, other parts of [ISO 13584](#) and other International Standards for representing supplier libraries for the purpose of exchange.

4.1 Generic resources

The generic resources consist of the following EXPRESS schemas:

- **ISO13584_instance_resource_schema**,
- **ISO13584_library_expressions_schema**,
- **ISO13584_table_resource_schema**,
- **ISO13584_variable_semantics_schema**,
- **ISO13584_domain_resource_schema**.

These schemas provide resource constructs that are generic in nature. They may be used outside [ISO 13584](#), and particularly in all the applications that use a data dictionary conformant with the ISO/IEC dictionary schema specified in [IEC 61360-2](#) and whose content is duplicated in an informative annex of [ISO 13584-42](#).

4.1.1 ISO13584_instance_resource_schema

The **ISO13584_instance_resource_schema** provides the resource constructs needed to describe instances of classes, or values of properties, whose corresponding data types are specified in accordance with the **ISO13584_ISO61360_dictionary_schema** specified in [IEC 61360-2](#) or with the **ISO13584_extended_dictionary_schema** specified in this part of [ISO 13584](#).

4.1.2 ISO13584_library_expressions_schema

The **ISO13584_library_expressions_schema** provides the resource constructs for representing expressions that evaluate to a value belonging to any of the data types defined in the **ISO13584_ISO61360_dictionary_schema**, specified in [IEC 61360-2](#) or in the **ISO13584_extended_dictionary_schema** specified in this part of [ISO 13584](#).

4.1.3 ISO13584_table_resource_schema

The **ISO13584_table_resource_schema** defines the set of resource constructs needed to describe tables and algebraic operations in tables. The algebraic operations in tables considered in the **ISO13584_table_resource_schema** include relational algebra, and set operations. This schema may be used to represent any kind of tables, regardless of whether they relate to the parts library application domain.

4.1.4 ISO13584_variable_semantics_schema

The **ISO13584_variable_semantics_schema** provides the resource constructs needed to reference the current values of the different elements that characterise an instance of a class, whether this class is modelled according to the **ISO13584_IEC61360_dictionary_schema**, or according to its extensions defined in the **ISO13584_extended_dictionary_schema**. Following the data model defined in [ISO 13584-20](#) for representing variables and their value-assignment mechanism, these resources are defined as subtypes of **variable_semantics**.

NOTE **variable_semantics** is defined in [ISO 13584-20](#).

4.1.5 ISO13584_domain_resource_schema

The **ISO13584_domain_resource_schema** provides the resource constructs needed to represent the set of allowed values that constitutes the domain of a variable in a given context. This set of allowed values may be independent of any other variables, or it may depend on the values of some other variables. The resource constructs introduced in this schema enable the characterisation of both kinds of domains. These resources are generic in nature and can be used for various purposes and in different application contexts. In this part of [ISO 13584](#), these resources are used in the **ISO13584_library_content_schema** to define the extension of a class.

4.2 Parts library specific resources

The parts library specific resources consist of the following EXPRESS schemas:

- **ISO13584_extended_dictionary_schema**,
- **ISO13584_library_content_schema**,
- **ISO13584_external_file_schema**,
- **ISO13584_method_schema**.

These schemas provide resource constructs that are specific to the parts library application domain. They enable the representation, within a supplier library, of hierarchies of item classes and representations for a library of those item classes.

4.2.1 ISO13584_extended_dictionary_schema

The **ISO13584_extended_dictionary_schema** contains the extensions of the ISO/IEC dictionary schema required for parts library representing and exchange. These extensions include the following:

- representation of classes of features;
- representation of functional model classes and functional view classes;
- representation of properties of functional model classes and functional view classes; and
- association of tables and documents with classes defined in a supplier library.

4.2.2 ISO13584_library_content_schema

In a dictionary, classes are intentionally defined. Their sets of possible instances are neither explicitly nor implicitly specified. The role of the **ISO13584_library_content_schema** is to enable the representation of class extensions. The set of possible instances is either explicitly described using a simple set structure, or implicitly defined by using the resources of the **ISO13584_domain_resource_schema**.

4.2.3 ISO13584_external_file_schema

The **ISO13584_external_file_schema** defines a mechanism and provides the resource constructs for referencing external files, whether these external files conform to an EXPRESS information model, and whether their file formats conform to an implementation method defined in [ISO 10303](#). The **ISO13584_external_file_schema** includes, in particular, all the material needed to support Internet-oriented documents and access mechanisms in a supplier library.

4.2.4 ISO13584_method_schema

The **ISO13584_method_schema** provides the resource constructs for representing methods for creating library item representations in a user modelling system.

4.3 Library integrated information models

A library integrated information model is an EXPRESS schema that integrates resource constructs from the above schemas and from other parts of [ISO 13584](#) or other International Standards for representing supplier libraries for the purpose of exchange. Library integrated information models are associated with conformance requirements. The library integrated information models defined in this part of [ISO 13584](#) enable the exchange of three kinds of libraries between a library data supplier and a library end-user:

- libraries that consist of hierarchies of classes of parts, materials or features,

- libraries that consist of representations of parts, materials or features, and
- libraries that define new representation categories capable of being provided for any family of parts, materials or features.

NOTE Each library integrated information model defined in this part of [ISO 13584](#) allows the exchange of only one of the above three kinds of libraries. However, due to the dictionary mechanism, these three kinds of libraries may be exchanged separately as three different supplier libraries, and then, these three different supplier libraries might be integrated in the same user library.

4.3.1 ISO13584_g_m_iim_schema and LIIM 24-1

The **ISO13584_g_m_iim_schema** specifies the information requirements for exchanging hierarchies of item classes, where items may be parts, materials or features. This schema is associated with a set of standard data, that defines the formats of library external files that may be referenced by **ISO13584_g_m_iim_schema** entity data type instances, and with implementation methods for the library delivery file. Together with the standard data specified in [annex E](#) and with the implementation methods specified in [annex F](#), the **ISO13584_g_m_iim_schema** constitutes the library integrated information model LIIM 24-1. Conformance requirements to LIIM 24-1 are defined in [clause 16](#) of this part of [ISO 13584](#).

4.3.2 ISO13584_f_m_iim_schema and LIIM 24-2

The **ISO13584_f_m_iim_schema** specifies the information requirements for exchanging hierarchies of library representations of library-defined item classes. This schema is associated with a set of standard data, that defines the formats of library external files that may be referenced by **ISO13584_f_m_iim_schema** entity data type instances, and with implementation methods for the library delivery file. Together with the standard data specified in [annex I](#) and with the implementation methods specified in [annex J](#), the **ISO13584_f_m_iim_schema** constitutes the library integrated information model LIIM 24-2. Conformance requirements to LIIM 24-2 are defined in [clause 17](#) of this part of [ISO 13584](#).

4.3.3 ISO13584_f_v_iim_schema and LIIM 24-3

The **ISO13584_f_v_iim_schema** specifies the information requirements for exchanging hierarchies of classes that define new representation categories capable of being provided for any library-defined item classes. This schema is associated with a set of standard data, that defines the formats of library external files that may be referenced by **ISO13584_f_v_iim_schema** entity data type instances, and with implementation methods for the library delivery file. Together with the standard data specified in [annex M](#) and with the implementation methods specified in [annex N](#), the **ISO13584_f_v_iim_schema** constitutes the library integrated information model LIIM 24-3. Conformance requirements to LIIM 24-3 are defined in [clause 18](#) of this part of [ISO 13584](#).

NOTE 1 Classes that define representation categories are functional view classes. Functional view classes are defined in ISO 13584-1.

NOTE 2 Only supplier-defined functional view classes need to be exchanged using the library integrated information model LIIM 24-3. Functional view classes defined by view exchange protocols are already stored in the dictionary of any implementation that claims conformance to these view exchange protocols.

5 Fundamental concepts and assumptions

The following concepts and assumptions apply to this part of [ISO 13584](#).

5.1 Conceptual model of a supplier library

The conceptual model of a supplier library is described in Part 10 of [ISO 13584](#). This conceptual model uses the object oriented paradigm for representing as a class:

- a family of parts; such a class is called a general model class,

- the representations of the different parts of a parts family belonging to the same representation category; such a class is called a functional model class, and
- the definitions of the different representation categories capable of being provided for any family of parts; such a class is called a functional view class.

These classes are structured hierarchically following the simple inheritance relationship.

This part of [ISO 13584](#) provides the EXPRESS resource constructs for representing such a hierarchy of classes for the purpose of exchange.

5.2 Implicit versus explicit description of a parts library

At the lower level of the hierarchy of general model classes are represented simple families of parts. Two approaches may be used for representing simple families of parts.

NOTE a simple family of parts is a set of parts where each part may be described by the same group of properties (see [clause 3](#)).

5.2.1 Explicit modelling of simple families of parts: by set extension

Explicit description consists of representing separately each part of the family, and gathering all these descriptions within a set structure.

EXAMPLE Let's assume a family F is to be captured. It is described by part characteristics a , b and c , and contains 90 different parts defined by:

- $a = 1, 2, \dots, 10$;
- $b = 1, 2, \dots, 10$;
- $c = (a + b) / 2$, and;
- there exists no part such that $a = b$.

A simplified information model stated in EXPRESS is presented in [Figure 1](#). Following this information model, the family F might be described, using the [ISO 10303-21](#) implementation method, as presented in [Figure 2](#).

```
ENTITY dic_part;  
    family_name: STRING;  
    properties: LIST [1:?] OF property_value;  
END_ENTITY;  
  
ENTITY property_value;  
    property_name: STRING;  
    value: REAL;  
END_ENTITY;
```

Figure 1 — Simplified example of an explicit information model for families of parts

```
#10=DIC_PART('F', (#100, #101, #102));  
    #100=PROPERTY_VALUE('a', 1.0);  
    #101=PROPERTY_VALUE('b', 2.0);  
    #102=PROPERTY_VALUE('c', 1.5);  
...  
#90=DIC_PART('F', (#900, #901, #902));  
    #900=PROPERTY_VALUE('a', 10.0);  
    #901=PROPERTY_VALUE('b', 9.0);  
    #902=PROPERTY_VALUE('c', 9.5);
```

Figure 2 — Example of explicit description of a family of parts

The set of instances is defined by set extension: every instance of the set is explicitly defined.

NOTE As shown by the above example the description and the data management of explicitly described families is rather simple provided that the number of instances is not too large.

5.2.2 Implicit modeling of simple families by entity data type

Using an entity-data-type-oriented information modeling language, implicit description consists of capturing all the set of instances in an entity data type, using the capabilities of the modeling language to restrict the implicit content of this set of instances.

EXAMPLE Using the EXPRESS language, the family *F* defined in the previous example might be modeled according to [Figure 3](#). Such an information model implicitly describes that there exist 90 value-different instances of the *F* family. The set of instances is defined by intention.

```
ENTITY F;
  a: REAL;
  b: REAL;
DERIVE
  c: REAL := (a + b) / 2;
WHERE
  WR1: SELF.a IN
    [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0];
  WR2: SELF.b IN
    [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0];
  WR3: SELF.a <> SELF.b;
END_ENTITY;
```

Figure 3 — Example of implicit description of a parts family in the EXPRESS language

NOTE From the user perspective, an implicitly-described parts family is a black-box where the user puts in some parameters values and gets back a set of part instances that conform to those values. Within the black-box, the family may be represented using tables, query functions, expressions, etc.

5.2.3 Explicit and implicit description of classes in this part of ISO 13584

This part of [ISO 13584](#) lets the library data suppliers choose which approach they want to use for describing the content of their families. A family is always described by a class, but the set of all instances satisfying a class definition may be represented either explicitly, or implicitly.

Advantages of the implicit approach for family representation are:

- Storage reduction for families that contain a large number of parts.

EXAMPLE 1 More than 5 billion of different screws conforming to ISO screw standards exist.

- The capability to model families of customisable parts where some properties may have any value in some continuous ranges.

EXAMPLE 2 In the catalogue of a linear bearings supplier, the length of the guide way of a linear bearing may be ordered for any value within some continuous range.

- The capability to model not only the characteristics of the parts, but also the context parameters whose values do not belong to part definition but that enable a part to be selected through the requirements that must be fulfilled for it to meet the user's needs.

EXAMPLE 3 Assume that, in a family of parts called *F*, there exists a context parameter called *requirement*, that may take any integer value between 1 and 9. Assume that, for each value, there exists only one part that fulfil the corresponding requirements, the part whose attribute *a* equals "requirement -1", and whose attribute *b* equals "requirement +1". The corresponding constraints may be modelled in EXPRESS. The family *F* might be modelled as shown in [Figure 4](#). Using such a description, a part of the family may be selected either through the *a* or *b* characteristics, or through the *requirement* context parameter.

```
ENTITY F;  
  requirement: INTEGER;  
  a: REAL;  
  b: REAL;  
WHERE  
  WR1: SELF.a IN  
    [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0];  
  WR2: SELF.b IN  
    [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0];  
  WR3: SELF.a <> SELF.b;  
  WR4: {1 < SELF.requirement <= 9};  
  WR5: (SELF.a = SELF.requirement - 1.0) AND  
    (SELF.b = SELF.requirement + 1.0);  
END_ENTITY;
```

Figure 4 — Capturing context parameters in an implicit description

NOTE In [Figure 4](#), the constraints are directly defined in EXPRESS. They cannot be exchanged using any of the implementation methods defined in [ISO 10303](#). The **ISO13584_domain_resource_schema** documented in [clause 10](#) of this part of [ISO 13584](#) provides EXPRESS resource constructs that permit implementations to exchange such constraints.

Advantages of the explicit approach for family representing include:

- Easiness of description of a library by the library data supplier;
- Easiness of implementation of LMS that support only explicit description of class extension;
- Easiness of modeling assemblies when the number of possible configurations is rather small;
- Easiness of splitting the properties associated with a part in a general model and various functional models and of gathering them again during user access.

5.3 Direct use of EXPRESS versus meta-modelling for implicit description

Two approaches may be used for implicitly representing families of parts using the EXPRESS language.

5.3.1 Direct use of the EXPRESS language for modelling classes

The use of a direct EXPRESS model of a class, as shown in [Figures 3](#) and [4](#) has the following advantages:

- the notion of a class in this International Standard may be mapped onto the EXPRESS notion of an entity data type;
- some required aspects of classes are enforced by the syntax of the EXPRESS language;

EXAMPLE 1 The constraint that an attribute shall be followed by a unique data type is a syntactic constraint.

- other required aspects of classes are enforced by the semantics of the EXPRESS language;

EXAMPLE 2 The constraint that only inherited or defined attributes may be referred to in the description of an entity data type is a semantic constraint.

- a mapping of a population of an EXPRESS model to an external file format exists.

EXAMPLE 3 The mapping defined in [ISO 10303-21](#) is such a mapping.

Directly modelling classes in EXPRESS has the following disadvantages:

- classes and instances are modelled and processed completely differently;

EXAMPLE 4 An EXPRESS schema is represented in ASCII and is intended to be compiled. A set of instances is represented, e. g., according to [ISO 10303-21](#) and is intended to be read by a STEP processor.

- only those concepts that exist in the EXPRESS language may be used in a class description.

EXAMPLE 5 In EXPRESS, it is not possible to capture the fact that the *requirement* attribute in [Figure 4](#) is semantically different from the attributes *a* and *b*. The value of the *requirement* attribute may be changed without changing the part, whereas changing either *a* or *b* would change the part.

5.3.2 Meta-modelling of classes using EXPRESS

A meta-model is a model that provides the ability to represent other models.

Using the meta-modelling approach, a class is represented as a set of instances of the EXPRESS entities belonging to a meta-model.

[Figure 5](#) shows a simple meta-model of a part class. The actual information model of a part class is located in the **ISO13584_library_content_schema** in [clause 12](#). The class shown in [Figure 4](#) directly modelled in EXPRESS might be represented in a similar way as shown in [Figure 6](#) using the meta-model of [Figure 5](#). Note that the actual information model of constraints is located in the **ISO13584_domain_resource_schema** in [clause 10](#).

This part of [ISO 13584](#) uses a meta-modelling approach for representing the three kinds of classes described in [clause 5.1](#).

NOTE As shown in [Figure 1](#), another information model may be defined for representing instances. Therefore, class and instance descriptions may be modelled at the same level and exchanged within the same population of EXPRESS entity data type instances. The **ISO13584_instance_resource_schema**, documented in [clause 6](#), provides EXPRESS resource constructs for representing class instances.

```
...
ENTITY part_class;
  characteristics: SET [0:?] OF property_definition;
  domain: SET [0:?] OF constraint;
END_ENTITY;

ENTITY property_definition;
  property_name: STRING;
  value_domain: data_type_specification;
END_ENTITY;

ENTITY data_type_specification;
  type_name: STRING;
END_ENTITY;
...
```

Figure 5 — Simple meta-model of a part class in EXPRESS

```
...
#10=PART_CLASS((#20,#30,#40),(#50,#60,#70,#80,#90));
#20=PROPERTY_DEFINITION("a", #120);
#30=PROPERTY_DEFINITION("b", #120);
#40=PROPERTY_DEFINITION("requirement", #120);
#50= ... /* constraint: a is in [1.0, 2.0, ..., 10.0] */
#60= ... /* constraint: b is in [1.0, 2.0, ..., 10.0] */
#70=CONSTRAINT(.NOT_EQUAL.,#20,#30); /* SELF.a <> SELF.b */
#80=CONSTRAINT(.EQUAL., #20, #100); /* SELF.a = value of #100 */
#90= ... /* constraint that b = SELF.requirement + 1.0 */
#100=LESS_EXPRESSION(#40, #110); /* SELF.requirement - 1.0 */
#110=REAL_VALUE(1.0);
#120=DATA_TYPE_SPECIFICATION('REAL_TYPE');
...
```

Figure 6 — Model of a part family using a meta-modelling approach

5.4 Two level description of a supplier library and the ISO/IEC common dictionary schema

Each parts family that constitutes a supplier library has two levels of description:

- The first level describes the concepts about the parts family. Each parts family is both informally defined by a natural language description and formally defined by a set of attributes. This first level constitutes the dictionary definition of the corresponding parts family. It is represented as a **dictionary_element**. A **dictionary_element** may be exchanged. It may also be stored in the semantic dictionary of the receiving system.

- The second level describes the set of parts belonging to the family. This second level constitutes the library specification of the corresponding parts family. It is represented as a **content_item** intended to be stored in a user library.

The **dictionary_element** describes only the abstract concept to which the parts family corresponds. It provides the information that determines whether a part may belong to that parts family. It does not define the set of parts that constitute the parts family.

The **content_item**, when present, completes the above description by specifying all the parts belonging to the parts family.

A large degree of independence separates these two levels of description. First, some parts families have only a dictionary element, and not a formally defined content. This is the case for the abstract parts families described in the terminology or dictionary standards.

EXAMPLE [IEC 61360-4](#) is an example of dictionary standard. This Standard describes, through a hierarchical structure, many families of parts that exist in electronics.

Secondly, the extension of a class, modelled by **content_item**, may change without altering the dictionary elements that apply to this class. This is the case when a product standard or a catalogue is updated by adding or withdrawing some parts of a parts family.

This part of [ISO 13584](#) provides for representing either the first level, or both the first and second levels:

- a) Parts libraries that contain only **dictionary_elements** are modelled through the **dictionary** entity defined in the **ISO13584_extended_dictionary_schema**. They correspond to conformance classes 1, 2 and 3 of library integrated information model 24-1 specified in [clause 16](#) of this part of [ISO 13584](#).
- b) Parts libraries that contain both **dictionary_elements** and **content_items** are modelled through the **library** entity defined in the **ISO13584_library_content_schema**. They correspond to conformance class 4, 5 and 6 of library integrated information model 24-1 specified in [clause 16](#) of this part of [ISO 13584](#).

5.4.1 Common dictionary description for ISO 13584 and IEC 61360

The **ISO13584_IEC61360_dictionary_schema**, documented in an informative annex of ISO 13584-42: 1998, is a common resource for [ISO 13584](#) and for [IEC 61360](#). This schema has been defined by ISO TC 184/SC4/WG2 and IEC SC3D for the exchange of those **dictionary_elements** that address their common scope. This scope consists of families of parts linked by the *is-a* relationship and described by a set of properties.

5.4.2 Dictionary descriptions for ISO 13584

The resource constructs needed to represent the **dictionary_elements** that are in scope for [ISO 13584](#) and out of scope for [IEC 61360](#), are provided in the **ISO13584_extended_dictionary_schema** defined in [clause 11](#) of this part of [ISO 13584](#). The scope of the **ISO13584_extended_dictionary_schema** is an extension of the **ISO13584_IEC61360_dictionary_schema**. It references all the resources defined in the **ISO13584_IEC61360_dictionary_schema**.

NOTE The **dictionary_elements** that are in scope for [ISO 13584](#) and out of scope for [IEC 61360](#) include: representation of functional model classes, representation of functional view classes, aggregation relationships.

5.4.3 Interoperability of ISO 13584 and IEC 61360

This part of [ISO 13584](#) specifies a number of options that may be supported by an implementation. These options have been grouped into conformance classes. Conformance requirements are defined in [clause 15](#) of this part of [ISO 13584](#). In particular, conformance to a particular conformance class requires that all entities, types, and associated constraints defined as part of that conformance class be supported.

Conformance class 0 of the library integrated information model 24-1, documented in [clause 16](#) of this part of [ISO 13584](#), only requires an implementation to support the common ISO and IEC requirements defined in the **ISO13584_IEC61360_dictionary_schema**. Thus, an [ISO 13584](#) implementation that conforms to that conformance class is able to handle any instance population of the ISO/IEC common **ISO13584_IEC61360_dictionary_schema**, using an implementation method defined for conformance class 0. Moreover, any conformance class of the library integrated information model 24-1 including the requirements defined in conformance class 0, is able to do the same.

NOTE The achievement of such an interoperability has introduced more complexity in both the ISO/IEC common **ISO13584_IEC61360_dictionary_schema**, that provides hooks for the extensions needed for [ISO 13584](#), and in the **ISO13584_extended_dictionary_schema**.

5.5 Independence between dictionary_elements and content_items: the BSU mechanism

In the [ISO 13584](#) Standard series, each piece of information intended to constitute a dictionary entry is represented through three entities:

- a) the **basic_semantic_unit** (BSU) entity carries the universal identification of this piece of information;
- b) the **dictionary_element** entity contains the set of attributes that constitutes the dictionary description of this piece of information;

EXAMPLE 1 Name, definition and type of value are examples of attributes that constitute the dictionary description of a piece of information.

- c) the **content_item** entity represents the possible values of this piece of information.

EXAMPLE 2 The **content_item** of a class, represented by the **class_extension** entity, specifies the possible instance values of this class.

NOTE In [ISO 13584](#), the pieces of information intended to constitute dictionary entries are: supplier, class, property, program library, type, table and document.

5.5.1 Reference between several EXPRESS schema populations via the BSU mechanism

All the references between dictionary entries shall be done through their **BSUs**. Thus, when a dictionary entry is intended to be referenced in some population, only its **BSU** entity instance is requested to belong to this population. Reference to a **BSU** stands for a reference to the complete piece of information that constitute a dictionary entry, whether its **dictionary_element** and **content_item** entities belong to the same population. Thus, this three levels model ensures independence between the **dictionary_element** entity instance, the **content_item** entity instance, and the **BSU** entity instance for any dictionary entry. This mechanism also provides a means to implement references between dictionary entries belonging to different EXPRESS schema populations, possibly stored in different exchange files or data repositories.

EXAMPLE 1 The **content_item** of a class may reference the properties that characterise the instances of the class through their **BSUs**. If the **dictionary_elements** that describe those properties are assumed to be available on the receiving system, the sending system may decide not to send these **dictionary_elements** together with the **content_items** that reference them.

EXAMPLE 2 In product model data, a product may be associated with a particular value for some property. When this property is modeled according to the **ISO13584_IEC61360_dictionary_schema**, only the **BSU** of this property needs to be represented in product data.

5.5.2 Expressing constraints between dictionary entries

All the constraints between **dictionary_elements** and/or **content_items** of dictionary entries are expressed by means of references to their **BSUs**. But these **dictionary_elements** and/or **content_items** may not be available in the exchange context where the **BSUs** are available. Thus, when the role of a

constraint is to restrict the allowed set of values of an attribute of either a **dictionary_element** or a **content_item**, the constraint may only be checked when and where the corresponding instance is available, otherwise the constraint shall be ignored. This behaviour is specified, at the EXPRESS code level, either by using the **definition_available_implies** function, for **dictionary_element** availability, or by checking the size of the aggregate value of the **referenced_by** attribute of a **basic_semantic_unit**, for **content_item** availability.

NOTE **definition_available_implies** and **basic_semantic_unit** are defined in the **ISO13584_IEC61360_dictionary_schema** specified in [IEC 61360-2](#), and duplicated for convenience in informative annex D of [ISO 13584-42](#).

5.6 ISO 13584 and the Internet

5.6.1 Documents represented within a library exchange context

A library exchange context includes a library delivery file conforming to one conformance class of a library integrated information model and several library external files. These library external files are referenced from the library delivery file using **external_item** entities. Such entities specify the content of the external files, the protocol that shall be used to process them and the external file addresses that identify the various library external files within the library exchange context.

This part of [ISO 13584](#) includes provisions for exchanging documents as library external files.

5.6.2 Support of the HTTP protocol and local Internet server

One particular network protocol for accessing documents is the HTTP protocol. This protocol provides for accessing documents whose contents are represented by means of several files, possibly encoded according to different file formats defined as MIME-like file, and possibly referencing other Internet resources by means of HTTP URLs.

This part of [ISO 13584](#) includes provisions for exchanging documents by means of library external files intended to be accessed using the HTTP protocol. Such library external files are called http files. Only http files whose values of MIME-defined fields of information registered by IANA are allowed for use in the context of this part of [ISO 13584](#).

When a conformance class of a LIIM or a conformance class of a VEP specifies that library external files according to the HTTP protocol shall be supported, an implementation that claims conformance to this conformance class shall include:

- a) the capabilities of an Internet server, as defined in IAB RFC 2068, for the purpose of storing documents;
- b) the capabilities of an Internet client, as defined in IAB RFC 2068, and the capabilities defined in IAB RFC 2049 for the purpose of recognising and browsing these documents; and
- c) a data repository, and its management system, to store and retrieve the data exchanged in the library delivery file.

The mechanisms defined in [clause 13](#) of this part of [ISO 13584](#) ensure that the hyperlink references specified by the library data supplier remain valid on the local Internet server. They also enable the library end-user to switch from document navigation to data retrieval and selection.

5.6.3 Particular HTTP formats to be supported by an implementation

In the context of the HTTP protocol, files exchanged over the Internet are MIME-like files. MIME defines the format of an exchanged file by means of several fields of information, called header fields and parameters, that are supposed to be added in the header of the files by the HTTP server.

In the context of this part of [ISO 13584](#), the values of MIME-defined fields of information that characterise the format of the http files that are exchanged as a library external files are represented explicitly in

the **http_file** entity that references them. When the supplier library is compiled on the receiving system site, these values are supposed to be used both to filter the files that the implementation intends to record, and to initiate the receiving site Internet server. These values are not duplicated in any header part of the library external files.

This part of [ISO 13584](#) uses the following MIME-defined header fields and parameters to specify the format of a file:

NOTE 1 These MIME-defined header fields and parameters are documented in IAB RFC 2045 and IAB RFC 2184.

- *MIME-Version* that defines the version of MIME: the value of this attribute is implicit, and it equals 1.0;
- *Content-Type* that defines the media type and subtype of data in the file;

EXAMPLE 1 "image" is an example of media type; "jpeg" is an example of subtype.

- *Content-Transfer-Encoding* that defines the encoding, or lack of encoding, of the file data for the purpose of exchange in the library exchange context.

EXAMPLE 2 "base64" is an example of data encoding defined in IAB RFC 2045.

- *Character set* parameter that specifies a method of converting a sequence of octets into a sequence of characters;

EXAMPLE 3 "ISO-8859-1" is an example of MIME "character set". The corresponding method of converting a sequence of bytes into a sequence of characters is defined in [ISO 8859-1](#).

NOTE 2 [ISO 8859-1](#) defines an 8-bit character encoding that includes most of the specific characters of western languages.

- *Language* parameter that specifies, either implicitly or explicitly, the language of a document.

This part of [ISO 13584](#) does not specify the formats that are allowed for use for the http files belonging to a library exchange context. It only specifies that:

- only values of MIME header fields and parameters registered by IANA are allowed;
- the local Internet client defined in the previous clause shall meet, at the minimum, the MIME conformance criteria defined in IAB RFC 2049 for the purpose of recognising and browsing http files;
- the local Internet client defined in the previous clause shall recognise "text" documents conforming to HTML as defined by IAB RFC 1866.

Apart from private agreement between the sender and the receiver, this part of [ISO 13584](#) also strongly recommends to restrict to a small number of formats:

- that are mature;
- that are stable;
- that are unambiguously characterised by MIME Content-Type;
- whose specifications are publicly available, or that are associated with public domain Internet-available readers.

5.6.4 Remote access to a document through the Internet

This part of [ISO 13584](#) contains provisions that enable a library data supplier to deliver documents within a library exchange context. It also contains provisions for providing a location where a document may be found by means of an absolute URL. Remote access to such documents is not required for the implementation that claims conformance to any library integrated information model defined in this part of [ISO 13584](#).

6 ISO13584_instance_resource_schema

This clause defines the requirements for the **ISO13584_instance_resource_schema**. The following EXPRESS declaration introduces the **ISO13584_instance_resource_schema** block and identifies the necessary external references.

EXPRESS specification:

```
*) SCHEMA ISO13584_instance_resource_schema; REFERENCE FROM ISO13584_IEC61360_
dictionary_schema (all_class_descriptions_reachable, basic_semantic_unit, class,
class_BSU, condition_det, content_item, data_type_BSU, definition_available_implies,
dependent_p_det, is_subclass, level, list_to_set, non_quantitative_code_type, non_
quantitative_int_type, property_BSU, version_type); REFERENCE FROM ISO13584_IEC61360_
language_resource_schema (translatable_label, present_translations); REFERENCE FROM
ISO13584_extended_dictionary_schema (a_priori_semantic_relationship, abstract_functional_
model_class, applicable_properties, data_type_class_of, data_type_level_spec, data_
type_level_value_typeof, data_type_type_name, data_type_typeof, functional_view_v_c_v,
data_type_non_quantitative_code_type, data_type_non_quantitative_int_type); REFERENCE
FROM ISO13584_library_content_schema (allowed_properties, explicit_functional_model_
class_extension, explicit_item_class_extension, functional_model_class_extension, fm_
free_model_properties_list, item_class_extension, gm_identification_characteristics_list,
method_variables, selectable_properties_list); REFERENCE FROM ISO13584_external_file_
schema (program_reference, representation_reference, property_value_external_item);
REFERENCE FROM geometry_schema (axis1_placement, axis2_placement_2d, axis2_placement_3d,
placement); REFERENCE FROM representation_schema (representation, representation_context,
representation_item); REFERENCE FROM product_definition_schema (product, product_category,
product_definition, product_definition_formation); REFERENCE FROM product_property_
definition_schema (property_definition); USE FROM person_organization_schema (address,
organization, person, person_and_organization, personal_address, organizational_address );
USE FROM date_time_schema (date, date_and_time, local_time, calendar_date, ordinal_date,
week_of_year_and_day_date); REFERENCE FROM geometry_schema (geometric_representation_
context); (*
```

NOTE The schemas referenced above can be found in the following documents:

ISO13584_IEC61360_dictionary_schema

[IEC 61360-2](#)

(which is duplicated for convenience in informative annex D of [ISO 13584-42](#)),

ISO13584_IEC61360_language_resource_schema

[IEC 61360-2](#)

(which is duplicated for convenience in informative annex D of [ISO 13584-42](#)),

ISO13584_extended_dictionary_schema

This part of [ISO 13584](#),

ISO13584_library_content_schema

This part of [ISO 13584](#),

ISO13584_external_file_schema

This part of [ISO 13584](#).

geometry_schema

[ISO 10303-42](#),

representation_schema

[ISO 10303-43](#),

product_definition_schema

[ISO 10303-41](#),

product_property_definition_schema

[ISO 10303-41](#),

person_organization_schema

[ISO 10303-41](#),

date_time_schema

[ISO 10303-41](#),

6.1 Introduction to the ISO13584_instance_resource_schema

The **ISO13584_instance_resource_schema** defines the resources needed to describe instances of classes and values of properties whose data types are specified in accordance with this International Standard.

NOTE The approach followed in this schema is largely based on the SDAI approach for describing EXPRESS types and instances, which is specified in [ISO 10303-22 \[7\]](#).

The **ISO13584_instance_resource_schema** models:

- the description of instances of properties whose types are specified either by the **ISO13584_IEC61360_dictionary_schema** resource constructs or by the **ISO13584_extended_dictionary_schema** resource constructs;
- the description of instances of classes whose dictionary elements are specified either by **ISO13584_IEC61360_dictionary_schema** resource constructs or **ISO13584_extended_dictionary_schema** resource constructs;
- the description of instances of classes whose contents are specified by **ISO13584_library_content_schema** resource constructs;
- the mechanisms for computing the data type to which an instance belongs and for ensuring type checking.

The **ISO13584_instance_resource_schema** does not model:

- the specification of the internal representation of a created instance or a selected property in a user library during a user selection process.

6.2 Fundamental concepts and assumptions for the ISO13584_instance_resource_schema

Description of classes requires capabilities for representing values of properties and instances of classes.

EXAMPLE To represent the domain of a data type that is restricted by a constraint defined by a table, one needs to represent the various values contained in the table. These values are values of properties or instances of classes.

NOTE 1 A value of a property is an instance of a class when the data type of this property is a **class_instance_type**.

NOTE 2 **class_instance_type** is defined in the **ISO13584_IEC61360_dictionary_schema** documented in [IEC 61360-2](#) (which is duplicated for convenience in informative annex D of [ISO 13584-42](#)).

6.2.1 Two-fold description of classes and instance representation

In this International Standard, a class may be specified at two levels of abstraction, either in intention by means of dictionary data, or as a class extension, by means of the resource constructs defined in the **ISO13884_library_content_schema** contained in this part of [ISO 13584](#). [Figure 7](#) shows a planning model of this two-fold class description, with the associated instance descriptions.

A dictionary description, provided by a **class** entity, defines the properties that are applicable to the class and their data types. An instance of a class that is only described by a **class** entity is represented as a **dic_class_instance**. It may include any of these applicable properties, and each property may have any value conformant with its data type.

A library specification, represented as a **model_class_extension**, further specifies the class by defining:

- what subset of the applicable properties have their values provided in the library to describe the different instances;

EXAMPLE 1 In the **class** dictionary definition of a screw class, the "mass" property may be listed as an applicable property. Nevertheless, the library data supplier may decide not to provide the "mass" values for the various screws represented by the screw class.

- the allowed set of values for those properties provided;
- the subset of these properties that is needed to completely identify one instance in this class;
- how the other property values may be derived from the above properties (derivation functions).

NOTE 1 Derivation functions are used to specify how values of some properties may be computed from values of other properties.

An instance of such a class, represented either as a **lib_item_instance**, in the case of an **item_class**, or as a **lib_f_model_instance**, in the case of a **functional_model_class**, shall only be defined:

- by properties that are provided in the class extension description, and
- by property values belonging to the allowed set of values for the class instance.

Each instance of such a class shall be associated with all the properties that are needed to identify one instance within its class. When such an instance is a partially defined instance, some of these properties may be associated with no value.

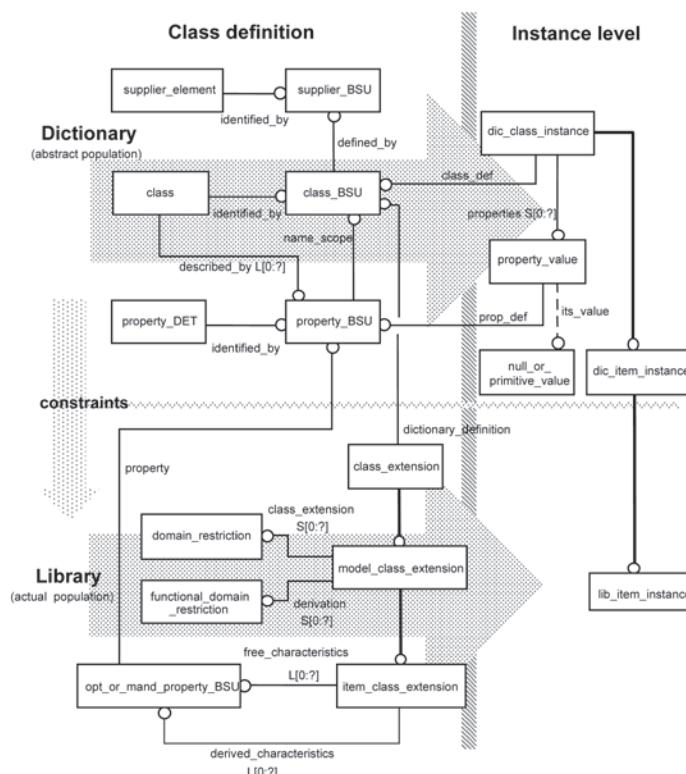


Figure 7 — Planning model of the relationships between class definition and the instance level

NOTE 2 Classes only specified by a dictionary element are able to represent standardised identification hierarchies intended to provide common semantics for computer interpretable data exchange. Classes also specified by a library specification are able to represent supplier catalogues or product standards.

NOTE 3 When a class is represented by an **item_class**, the properties needed to completely identify an instance are computed by the **gm_identification_characteristics_list** function. When a class corresponds with a **functional_model_class**, the properties needed to completely characterise an instance are computed by the **selectable_properties_list** function.

EXAMPLE 2 The component classes and data element types defined in [IEC 61360-4](#) describe the components and properties used in the electronic applications area. This description constitutes a set of **dictionary_elements** conformant to this International Standard. An instance of a component class defined in [IEC 61360-4](#) (for instance, a *fixed_linear_resistor*) may be modelled as a **dic_class_instance**.

NOTE 4 When a class is only defined through a dictionary element, such as the various classes defined in [IEC 61360-4](#), the properties applicable to this class are only associated with a data type. In an instance of such a class, each property may have whatever value belonging to its data type.

EXAMPLE 3 For describing its own parts catalogue, John Croke Inc. does not need only to capture the fact that the components produced in the factory are kind of "*light_dependent_resistors*", as defined in [IEC 61360-4](#). It also wants to describe which particular "*light_dependent_resistors*" the company is able to provide. This can be done by a library specification.

In the library specification, John Croke Inc describes the different *light_dependent_resistors* contained in its catalogue. It also specifies that the *resistance* is the only property needed to identify any resistor when such a resistor is ordered in its company. It may also provide tables that enable the system to generate the values of all the other properties he wants to provide for its *light_dependent_resistors* from the value of the *resistance*.

A resistor that is an instance of *light_dependent_resistors* class may be modeled as a **lib_component_instance** (a subtype of **lib_item_instance**). It will contain a **property_value** that references its *resistance* property. The values of the other properties of its *light_dependent_resistors* may also be associated with this **lib_component_instance**. In this case, these values should comply with the values computed by the derivation function specified in the **item_class_extension**.

NOTE 5 *resistor_noise_index* is an example of property of resistors whose definition is provided in [IEC 61360-4](#).

6.2.2 Representation of a context-dependent characteristic value

A context-dependent characteristic is a property whose value depends on some context parameter(s). If an instance is associated with a value of such a property, values shall also be provided for all the context parameters on which this value depends.

EXAMPLE The life span of a ball-bearing is a context-dependent characteristic. Its value depends on the radial load, the axial load and the rotational speed supported by that bearing. If an instance of a class that represents a ball-bearing family is associated with a value for the life span property, it shall also be associated with values for the radial load, the axial load and the rotational speed context parameters.

6.2.3 Optional properties

In the **ISO13584_library_content_schema**, a property of an item belonging to a library may be specified as optional. Such a property may be assigned a NULL value for some class instances. In the **ISO13584_instance_resource_schema**, the value of such a property is represented as an OPTIONAL value for the corresponding **property_value**.

NOTE This part of [ISO 13584](#) does not make any difference between a NULL value and an indeterminate value (?). In a **property_value**, a lack of value is represented as indeterminate. In a table **column**, that is a list, a lack of value is represented by a NULL value.

6.3 ISO13584_instance_resource_schema type definitions

This clause defines the types for modelling the values that may be assigned to properties in the **ISO13584_instance_resource_schema**.

6.3.1 Null_value

The **null_value** entity provides for the definition of data values which may be indeterminate value according to the ? value of the EXPRESS language.

EXPRESS specification:

```
*) ENTITY null_value; END_ENTITY; -- null_value (*)
```

6.3.2 Primitive_value

A **primitive_value** is any value that may be assigned to a property in the **ISO13584_extended_dictionary_schema**.

EXPRESS specification:

```
*) TYPE primitive_value = SELECT( simple_value, complex_value); END_TYPE; --  
primitive_value (*)
```

6.3.3 Null_or_primitive_value

A **null_or_primitive_value** is any value, included the **null_value**, that may be assigned to a property in the **ISO13584_extended_dictionary_schema**.

EXPRESS specification:

```
*) TYPE null_or_primitive_value = SELECT( null_value, primitive_value); END_TYPE; --  
null_or_primitive_value (*)
```

6.3.4 Simple_value

A **simple_value** is an unstructured value belonging to the EXPRESS NUMBER, STRING or BOOLEAN types. A STRING value may be represented in different languages.

NOTE 1 A **translatable_string_value** is considered as a **simple_value** as only one particular **string_value** is supposed to be displayed in any context

NOTE 2 The function that chooses in any context the particular value to be displayed shall be implementation dependent and is outside the scope of this part of [ISO 13584](#).

EXPRESS specification:

```
*) TYPE simple_value = SELECT( number_value, translatable_string_value, boolean_value);  
END_TYPE; -- simple_value (*)
```

6.3.5 Null_or_simple_value

A **null_or_simple_value** is an unstructured value belonging to the EXPRESS NUMBER, STRING or BOOLEAN types, included the **null_value**.

EXPRESS specification:

```
*) TYPE null_or_simple_value = SELECT( null_value, simple_value); END_TYPE; -- null_or_  
simple_value (*)
```

6.3.6 Number_value

A **number_value** is a value belonging to the EXPRESS NUMBER type.

EXPRESS specification:

```
*) TYPE number_value = SELECT( integer_value, real_value); END_TYPE; -- number_value (*)
```

6.3.7 Null_or_number_value

A **null_or_number_value** is a value belonging to the EXPRESS NUMBER type, included the **null_value**.

EXPRESS specification: