INTERNATIONAL STANDARD

ISO
13584-101

First edition
2003-05-15

Industrial automation systems and integration — Parts library —

Part 101:
Geometrical view exchange protocol by parametric program

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

Partie 101: Protocole d'échange de vues géométriques par programme paramétré

© ISO 2003

Contents

Figures

Tables

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 13584-101 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 ISO 13584 is described in ISO 13584-1. The numbering of the parts of ISO 13584 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 protocols.

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.

ISO 13584 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, conformance testing, view exchange protocol, and standardized content. The series are described in ISO 13584-1. This part of ISO 13584 is a member of the view exchange protocol series.

A view exchange protocol specifies how a particular representation category of the items described in a parts library may be exchanged in a library exchange context. It defines the identification of the representation category, the means to be used to exchange representations that belong to this representation category, the implementation resources that shall be available on any implementation that claims conformance to this view exchange protocol, and the standard data that shall be recognized by any implementation that claims conformance to this view exchange protocol.

This part of ISO 13584 specifies how geometric representations of the items described in a parts library may be exchanged by means of parametric FORTRAN programs based on the application programming interface specified in ISO 13584-31:1999, or, by case of separate agreement between the sender and the receiver, by means of non-standardized parametric formats.

Industrial automation systems and integration – Parts library – Part 101: Geometrical view exchange protocol by parametric program

1 Scope

This part of ISO 13584 specifies a representation category called *basic_geometry*. This representation category captures the generic concepts of the shape of a part. This representation category may be associated with any of the items defined in a parts library. This part of ISO 13584 also defines how representations that belong to this representation category may be exchanged within a library exchange context by means of FORTRAN programs compliant with ISO 13584-31:1999.

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

— the definition of the *basic_geometry* representation category and the mechanisms to be used to reference it;

— the properties to be used to characterize a particular representation within the *basic_geometry* representation category;

— the exchange format to be used for the library external files that describe the *basic_geometry* representations of classes of items described in a parts library by means of FORTRAN programs based on the application programming interface specified in ISO 13584-31:1999;

— the mechanism to be used, by case of separate agreement between the sender and the receiver, to reference external files that describe the *basic_geometry* representations of classes of items described in a parts library by means of formats not specified in ISO 13584;

— the implementation resources to be supported on any implementation that claims conformance to this part of ISO 13584;

— the dictionary entries to be supported by any implementation that claims conformance to this part of ISO 13584;

— the standard data to be recognized by any implementation that claims conformance to this part of ISO 13584.

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

— the structure and exchange format of a library delivery file that includes references to the representation category defined in this part of ISO 13584, and/or to the library external files whose exchange format is specified.

NOTE 1   The structure of a library delivery file is defined by a library integrated information model specified in one of the logical resource series parts of ISO 13584.

NOTE 2   The **ISO13584_f_m_iim_schema**, documented in ISO 13584-24, is a library integrated information model that defines the structure of a library delivery file. Such a library delivery file may contain instance values that reference the representation category and/or the library external files defined in this part 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/IEC 8824-1: 1998, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8859-1:1998*, Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

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

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

ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts*

ISO 10303-42:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation*

ISO/IEC 10646-1:2000, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*

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

ISO 13584-24: —[1), *Industrial automation systems and integration — Parts library — Part 24: Logical resource: Logical model of supplier library*

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 parts families*

---

1) To be published.

3 Terms, definitions and abbreviated terms

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

— ISO 10303-1:1994;

— ISO 10303-11:1994;

— ISO 10303-31:1994;

— ISO 10303-42:2000;

— ISO 13584-1:2001;

— ISO 13584-24:—[1);

— ISO 13584-31:1999;

— ISO 13584-42:1998.

3.1

application programming interface
API
set of functions that may be triggered by a program
[ISO 13584-24: —[1)]

3.2

basic semantic unit
entity that provides an absolute and universally unique identification of certain objects of the application domain
[ISO 13584-42:1998, definition 3.4.1]

3.3

binding
description of the concrete syntax that shall be used in a particular programming language to trigger the different functions that constitute an application programming interface
[ISO 13584-31:1999, definition 3.2.3]

3.4

computer aided design system
CAD system
kind of computer modelling system that generates and manages product data

3.5

conformance class
subset of a standard for which conformance may be claimed
[ISO 13584-24: —[1)]

---

1) To be published.

3.6

conformance requirement
precise, text definition of a characteristic required to be present in a conforming implementation
[ISO 10303-1:1994, definition 2.1.14]

3.7

conforming implementation
implementation which satisfies the conformance requirements defined by one or several
conformance classes of a standard
[ISO 13584-24: —[1]]

3.8

conformity; conformance
fulfilment by an implementation of all requirements specified
[ISO 10303-31:1994, definition 3.2.25]

3.9

constructive solid geometry
CSG
type of geometric modelling in which a solid is defined as the result of a sequence of regularized
Boolean operations operating on solid models
[ISO 10303-42:2000, definition 3.1.11]

3.10

entity data type instance
named unit of data which 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
[ISO 10303-11:1994, definition 3.2.7]

3.11

entity
class of information defined by common properties
[ISO 10303-11:1994, definition 3.2.5]

3.12

entity data type
representation of an entity. An entity data type establishes a domain of values defined by common
attributes and constraints
[ISO 10303-11:1994, definition 3.2.6]

3.13

functional model of a part
library data that represent one representation category of a part in an integrated library.
[ISO 13584-1:2001]

---

1) To be published.

EXAMPLE   A functional model of a precisely defined screw may consist of parametric programs which may be used to generate different geometric functional views of the screw in a CAD system database.

3.14

functional view of a part
data that represent one representation category of a part in product data
[ISO 13584-1:2001]

EXAMPLE   The structure of a functional view corresponding to geometry is not dependent on the part to be represented. This structure is specified as a functional view class.

3.15

implementation
software development in a given programming environment

3.16

implementation method
technique used by computer systems to exchange data that is described using the EXPRESS data specification language
[ISO 13584-24:—[1)]]

3.17

implementation resources
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
[ISO 13584-24: —[1)]]

3.18

information model
formal model of a bounded set of facts, concepts or instructions to meet a specified requirement
[ISO 10303-1:1994]

3.19

integrated library
operational system consisting of a Library Management System and a user library
[ISO 13584-1:2001]

3.20

library data supplier
organization that delivers a library in the standard format defined in ISO 13584 and is responsible for its content
[ISO 13584-1:2001]

---

1) To be published.

© ISO 2003 – All rights reserved

5

3.21

library delivery file
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
[ISO 13584-24:—[1)]]

> NOTE   A library delivery file specifies the structure and the content of a supplier library. It may reference library external files.

3.22

library end user
user of an integrated library who:

—   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
[ISO 13584-1:2001]

3.23

library exchange context
set of one library delivery file and zero, one or several library external files that represent together a supplier library
[ISO 13584-24:—[1)]]

3.24

library external file
file, referenced from a library delivery file, that contributes to the definition of a supplier library
[ISO 13584-24:—[1)]]

> NOTE   The structure and the format of a library external file is specified in the library delivery file that references it.

3.25

library integrated information model
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
[ISO 13584-24:—[1)]]

> NOTE   Three library integrated information models are defined in ISO 13584-24 for representing different kinds of supplier libraries.

---

1) To be published.

6

3.26

library management system
LMS
software system enabling the library end-user to use the content of an integrated library
[ISO 13584-1:2001]

> NOTE   This software system is not standardized.

3.27

parts library
identified set of data and possibly programs which may generate information about a set of parts
[ISO 13584-1:2001]

3.28

reference coordinate system
underlying global rectangular Cartesian coordinate system to which all geometry refers

3.29

representation category
abstraction used to distinguish between various possible user requirements regarding a part
representation
[ISO 13584-1:2001]

> NOTE   In the model defined in this International Standard, this distinction is formally expressed in terms
> of a view logical name and in terms of the view control variables.

3.30

standard data
requirement on a software system defined by means of EXPRESS entity data type instances that are
supposed to be recognized by this software system
[ISO 13584-24:—[1)]]

3.31

supplier library
set of data, and possibly of programs, for which the supplier is identified and that describes in the
standard format defined in ISO 13584 a set of parts and/or a set of representations of parts
[ISO 13584-1:2001

3.32

user library
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
[ISO 13584-1:2001]

[1)] To be published.

7

3.33

view control variable
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
[ISO 13584-1:2001]

> EXAMPLE   For the geometry perspective of a part, such view control variables values could be 2D, wire frame or solid.

3.34

view exchange protocol
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
[ISO 13584-24:—[1)]]

3.35

view logical name
identifier of an abstract representation category corresponding to a perspective that can be adopted by a user regarding a part
[ISO 13584-1:2001]

> EXAMPLE   Such identifiers could be geometry, inertia, kinematics, etc.

3.36 Abbreviated terms

For the purposes of this part of ISO 13584, the following abbreviated terms apply:

— 2D: Two Dimensional;

— 3D: Three Dimensional;

— API: Application Programming Interface;

— CAD: Computer Aided Design;

— CSG: Constructive Solid Geometry.

4 Identification of the *basic_geometry* representation

4.1 Concepts

When a library item is selected, it is often useful to access the shape of this item. Such a shape is a generic concept that may be captured at various levels of completeness and detail. These levels may be defined independently of the library items by the requirements that are intended to be met by each level.

To address this requirement, this part of ISO 13584 defines:

— a particular representation category, called *basic_geometry*, that provides for the generic concept of the shape of a library item;

---

1) To be published.

— five view control variables that are used to characterize the conceptual requirements that are intended to be met by each shape associated with the same library item;

— an exchange format of the *basic_geometry* representation category intended to be referenced from its library description and that consists of FORTRAN programs based on the ISO 13584-31:1999 standard API;

— a generic exchange mechanism that may be used by case of separate agreement between the sender and the receiver, and that provides the library description of this representation category by means of non-standardized parametric formats.

4.2 Standardized dictionary entries

The information model of the *basic_geometry* representation category shall be defined by standard data that consist of a functional view class entity data type instance. Such a functional view class is captured by an instance of the **functional_view_class** entity data type.

> NOTE 1   Functional view of a part is defined in ISO 13584-1. The **functional_view_class** EXPRESS entity data type is defined in ISO 13584-24.

> EXAMPLE   The information model of the *basic_geometry* representation category might be defined in EXPRESS through attributes like the geometry *level*, and a *content* that is an ISO 10303-43 **representation**. Such a description might be as follows:

```
ENTITY basic_geometry;
    level: geometry_level_type;
    ...
    content: representation;
END_ENTITY;
```

> Using the EXPRESS resource constructs defined in ISO 13584-24, the same information model may also be defined and exchanged as an instance of a **functional_view_class** entity data type, as follows:

```
#1 = FUNCTIONAL_VIEW_CLASS(..., 'basic_geometry', (#10, ...), (#11));
#10 = PROPERTY_BSU('level',...);/* a view control variable */
#11 = PROPERTY_BSU('content',...);   /* a view property */
```

> The latter description is the one used by this part of ISO 13584, and the corresponding instances will be found in the dictionary of a conforming implementation.

The universal identification of *basic_geometry* functional view class by means of a basic semantic unit is defined in 4.2.1. The universal identification of the view control variables defined for this functional view class, together with their domains of values are defined in 4.2.2. The **functional_view_class** instance shall not contain any **view_properties**. The attribute **its_superclass** of this instance does not exist, i.e., the corresponding functional view class has no superclass.

> NOTE 2   **view_properties** is an attribute of **functional_view_class** defined in ISO 13584-24. **its_superclass** is an attribute of **class** defined in ISO 13584-42.

4.2.1 View logical name

The basic semantic unit that identifies the functional view class that captures the representation category defined in this part of ISO 13584 contains the attributes shown in Table 1.

Table 1 — View logical name description

Class_code	Class_version	Supplier_code
'basic_geometry'	'001'	'0112/1///13584_101_1'

This set of information provides an universal identification of the *basic_geometry* representation category. It constitutes its view logical name.

NOTE   Subclauses D.3.3.4 and D.3.4.1 of ISO 13584-42:1998 require that the supplier code specified in ISO 13584-26 be used for any ISO 13584 data exchange.

4.2.2 View control variables

The functional view class is characterized using the view control variables whose universal identification and respective datatypes are described in Table 2 and illustrated in Figure 1.

4.3 Rules for the shapes to be provided in the *basic_geometry* representation category

This subclause defines the conceptual requirements that are intended to be met by each particular shape associated with a library item.

4.3.1 Geometry level

Geometry level 1: corresponds to a two-dimensional geometric representation that may be defined using the ISO 13584-31:1999 API. This geometry level defines a 2D representation shape that consists of points, lines and curves and contours, that may be opaque.

Geometry level 2: corresponds to a three-dimensional wireframe representation that may be defined using the ISO 13584-31:1999 API. This geometry level defines a 3D representation shape that consists of points, lines, and curves whose edges represent a shape.

Geometry level 3: corresponds to a three-dimensional solid representation that may be defined using the ISO 13584-31:1999 API. This geometry level defines a shape representation that consists of points, lines, curves and solids.

Table 2 — View control variables of the *basic_geometry* functional view class

Code	Version	Values	Corresponding labels (in English)
'geometry_level'	'001'	1 2 3	'2D' 'wireframe' 'solid'
'detail_level'	'001'	1 2 3	'simplified' 'standard' 'extended'
'side'	'001'	0 1 2 3 4 5 6	'null' 'front' 'rear' 'right' 'left' 'top' 'bottom'
'variant'	'001'	0 1 2 3 ... n	'null' 'external_shape' 'section' *reserved for future registration*
'unreg_variant'	'001'	0 1 ... n	'null' *library data supplier dependent*

4.3.2 Detail level

Detail level 1: the representation shall include, as a minimum, an indication of the reference coordinate system of the library item.

Detail level 2: the representation shall include, as a minimum, all the geometric representation items that are needed to establish a spatial relationship between the library item and the environment in which the item is intended for insertion.

Detail level 3: the representation shall include, as a minimum, all the geometric representation items that provide for overlapping (in 2D) or collision detection (in 3D) and it shall provide a representation that may be used in a detail design of the corresponding item.

4.3.3 Side

The meaning of the side view control variable and its relationship with the **geometric_representation_context** of the library item are illustrated in Figure 1.

This figure shows how the side is defined according to the definition of the reference coordinate system of the library item.
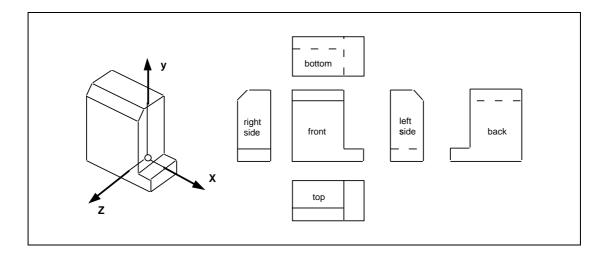
Figure 1 — Side view control variable meaning

NOTE   The side view control variable shall have a value equal to 0 when the geometry level view control variable defines a three-dimensional representation. This constraint on the corresponding view control variables **view_control_variable_range**s is formally expressed in the **ISO13584_101_side_and_geometry_level_compatibility_rule** included in the EXPRESS schema defined in Clause 6.3.2.1.

4.3.4 Variant

Variant 0: this variant level specifies that an unregistered variant is defined.

NOTE   Unregistered variant is defined in Clause 4.3.5.

Variant 1: this variant level specifies that the represented shape is the external shape of the library item.

Variant 2: this variant level specifies that the shape represented includes at least one cross-section.

Variant 3..n: these variant levels are reserved for future use by this part of ISO 13584.

NOTE   In the various conformance classes defined in this part, when defining a standardized geometric representation variant **view_control_variable_range**, variant 1 is mandatory. This constraint is formally expressed in the **ISO13584_101_variant_and_unregistered_variant_compatibility_rule** included in the EXPRESS schema defined in Clause 6.3.2.2.

4.3.5 Unregistered variant

Unregistered variant 0: this unregistered variant level specifies that a standardized variant is defined.

Unregistered variant 1..n: these unregistered variant levels specify library data supplier dependent variant levels.

NOTE   In the various conformance classes defined in this part, variant 0 is mutually exclusive with unregistered variant 0. This constraint is formally expressed at the level of the corresponding **view_control_variable_range**s in the **ISO13584_101_variant_and_unregistered_variant_compatibility_rule** included in the EXPRESS schema defined in Clause 6.3.2.2.

5 Exchange format

This clause defines the requirements for the library external files compliant with conformance classes 1, 2 and 3 of this part of ISO 13584. The structure and the format of the library external files compliant

with conformance classes 1E, 2E and 3E shall be defined by separate agreement between the sender and the receiver. The agreement shall specify the values of the attributes of the **program_protocol** entity instance that shall be used to characterize this structure and format.

5.1 FORTRAN SUBROUTINE name

The FORTRAN program unit shall be a SUBROUTINE.

> NOTE   The FORTRAN program unit may not be a PROGRAM or a FUNCTION.

The name of the FORTRAN program unit shall be built according to the following structure:

```
'F' + encoded_supplier_code + '_' + <program_name>
```

The `encoded_supplier_code` shall be the supplier code of the library data supplier, as defined by ISO 13584-26 on which an encoding function is applied. This encoding function, `encode`, is intended to replace each character that is not allowed in FORTRAN names by its corresponding encoded character that corresponds to the sequence "_" followed by the character's code (on exactly two characters) as defined in the ISO/IEC 10646-1 character set. Moreover, if the character to be encoded is "_", then the encoding result is "__".

The `encoded_supplier_code` is encoded using the following `encode` function:

```
*)
FUNCTION encode(s: STRING): STRING;

LOCAL
    strtmp: STRING;
    lower_cases: LIST OF STRING :=
        ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' ,'k', 'l', 'm',
         'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'];
    upper_cases: LIST OF STRING :=
        ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J' ,'K', 'L', 'M',
         'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'];
    numerals: LIST OF STRING :=
        ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'];
    underscore: STRING := '_';
    allowed_characters: LIST OF STRING :=
        lower_cases + upper_cases + numerals + underscore;
    result: STRING := '';
END_LOCAL;


REPEAT i := 1 TO LENGTH(s);
    IF NOT(s[i] IN allowed_characters)
    THEN
        strtmp := FORMAT(icode(s[i]), '02I');
        result := result + '_' + strtmp;
    ELSE
        IF (s[i] = '_')
        THEN
            result := result + '_' + s[i];
        ELSE
            result := result + s[i];
        END_IF;
    END_IF;
END_REPEAT;

RETURN (result);
END_FUNCTION;
(*
```

NOTE 1   Function `icode` returns the character code of a character in the ISO/IEC 10646-1:1993 character set, interpreted as an integer value.

NOTE 2   Function `icode` is not specified in EXPRESS because the limitations of the EXPRESS language would make such a function extremely long.

EXAMPLE   The 0112/1///13584_101_1 supplier code is encoded by 0112_2F1_2F_2F_2F13584__101__1. '2F' corresponds to the code of the '/' character in the ISO/IEC 10646-1:1993 character set.

The library data supplier shall use a supplier code that maps to a string of length equal to 31 characters or less.

NOTE 3   The string size resulting of the encoding process may exceed the allowed length for defining the FORTRAN SUBROUTINE name (limited to 31 characters). It is therefore the responsibility of the library data supplier to register a new supplier code in order to be able to build the FORTRAN SUBROUTINE name.

5.2 FORTRAN restrictions

5.2.1 Excluded statements

The following FORTRAN statements shall not be used:

— BACKSPACE,

— BLOCK DATA,

— CLOSE,

— COMMON,

— DATA,

— ENDFILE,

— ENTRY,

— EQUIVALENCE,

— FORMAT,

— INQUIRE,

— OPEN,

— PROGRAM,

— READ,

— REWIND,

— SAVE,

— STOP,

— WRITE.

NOTE   The FORTRAN program units within a parts library are intended to be run in various environments, on different CAD systems and following various operating techniques. In order to ensure maximal portability of these programs, the statements listed above are forbidden.

EXAMPLE   Compiling, linking, interpretation and translation are operating techniques.

5.2.2 Obsolete features

The following features have been declared obsolescent in FORTRAN. Therefore, the use of these features shall be avoided in new programs:

— Arithmetic-IF,

— Alternate return from subroutine,

— ASSIGN,

— Assigned FORMAT specifier,

— Assigned GOTO,

— DO loop control variables that are not integers,

— DO loop not ending on CONTINUE,

— Branch to END IF from outside IF block,

— H edit descriptor,

— PAUSE.

> NOTE   These features are still present, but will not appear in the next revision of ISO 1539:1991.

5.2.3 Exchange of a FORTRAN program unit

There shall be one and only one FORTRAN SUBROUTINE in one library external file that may be referenced to create a library item representation. Additional subroutines that are called from this SUBROUTINE may be present in the same library external file.

5.2.4 Character encoding

This library external file shall use only the 26 upper- and lower-case letters of the English alphabet, the 10 Arabic numerals, 0 to 9, the underscore, '_', and the special characters listed in Table 3 encoded according to ISO 8859-1.

Table 3 — The special characters of the FORTRAN language

Character	Name	Character	Name
=	Equals sign	:	Colon
+	Plus sign		Space
−	Hyphen-minus sign	!	Exclamation mark
*	Asterisk	"	Quotation mark
/	Solidus	%	Percent sign
(	Left parenthesis	&	Ampersand
)	Right parenthesis	;	Semicolon
,	Comma	<	Less-than sign
.	Full stop	>	Greater-than sign
$	Dollar sign	?	Question mark
'	Apostrophe		

5.3 Status of the program

The FORTRAN subroutine shall be in source code if the corresponding instance of the **library** entity data type is of type **library_in_standard_format**. It may be in source or object code otherwise.

NOTE **library** and **library_in_standard_format** are defined in ISO 13584-24.

6 Conformance requirements

This part of ISO 13584 defines six conformance classes.

For conformance classes 1 to 3, the format of the external file is defined in Clause 5. For conformance classes 1E to 3E, the format of the external file shall be defined in a separate agreement between the sender and the receiver.

An implementation claiming conformance to conformance class of this part of ISO 13584 must support the dictionary entries defined in Clause 4, and must recognize any data that obey the constraints defined in Clause 6.3.

An implementation claiming conformance to conformance class 1, 2 or 3 must also include the implementation resources specified in Clause 6.1, and must have the capability to process external files compliant with Clause 5.

An implementation claiming conformance to conformance class 1E, 2E or 3E must also include the implementation resources specified in Clause 6.1, and must have the capability to process external files associated with the **program_protocol** that the implementation claims to support.

NOTE The **program_protocol** is defined in ISO 13584-24.

6.1 Implementation resources

This clause gives requirements for a receiving system.

NOTE 1 For conformance classes 1, 2 and 3, the file format is the FORTRAN binding of ISO 13584-31:1999. For conformance classes 1E, 2E and 3E, the file format is to be defined by separate agreement between the sender and the receiver.

NOTE 2   The particular interface intended to be used to process a library external file is specified, within the library delivery file that references them, by means of an **external_file_protocol** entity.

The implementation resource requirements are the following:

—   Implementations claiming conformance to conformance class 1 shall support level 1 of the geometric programming interface API, as defined in Clause 6.1.1 of ISO 13584-31:1999. It corresponds to the creation of a *basic_geometry* representation for a *geometry_level* equal to 1.

—   Implementations claiming conformance to conformance class 2 shall support level 2 of the geometric programming interface API, as defined in Clause 6.1.1 of ISO 13584-31:1999. It corresponds to the creation of a *basic_geometry* representation for *geometry_level*s equal to 1 and 2.

—   Implementations claiming conformance to conformance class 3 shall support level 3 of the geometric programming interface API, as defined in Clause 6.1.1 of ISO 13584-31:1999. It corresponds to the creation of a *basic_geometry* representation for *geometry_level*s equal to 1, 2 and 3.

—   Conformance class 1E requires the support of a geometric interface able to create a *basic_geometry* representation for a *geometry_level* equal to 1.

—   Conformance class 2E requires the support of a geometric interface able to create a *basic_geometry* representation for *geometry_level*s equal to 1 and 2.

—   Conformance class 3E requires the support of a geometric interface able to create a *basic_geometry* representation for *geometry_level*s equal to 1, 2 and 3.

NOTE 3   New conformance classes are intended to be defined by amendment of this part of ISO 13584 if new APIs or new language bindings to the API in ISO 13584-31:1999 are defined.

6.2 Implementation methods

The implementation method for the library delivery file that references library external files compliant with this part of ISO 13584 shall be defined by the library integrated information model referenced by the library delivery file.

The implementation method for library external files compliant with conformance class 1, 2 and 3 of this part of ISO 13584 is specified by Clause 5. Implementation methods for the library external files compliant with conformance classes 1E, 2E and 3E shall be defined by separate agreement between the sender and the receiver.

NOTE   Identification of the library delivery file and the medium to be used for storing the library delivery file and the library external files referenced by this library delivery file are defined by separate agreement between the sender and the receiver and are outside the scope of this part of ISO 13584.

6.3 Constraints on a library delivery file for referencing this view exchange protocol

This subclause defines the **view_exchange_protocol_identification** instance values that are allowed for use in a library delivery file to reference the view exchange protocol defined in this part of ISO 13584.

NOTE 1   **view_exchange_protocol_identification** is defined in ISO 13584-24.

The set of allowed values is defined in Table 4 and in three EXPRESS schemas. Table 4 gives the allowable values of **view_exchange_protocol_identification.name** and **view_exchange_protocol_identification.application** for each conformance class.

The **ISO13584_101_cc_1_or_2_or_3_schema** schema and the **ISO13584_101_cc_1E_or_2E_or_3E_schema** schema contain a global rule, the role of which being

to specify the allowed values for the other attributes of **view_exchange_protocol_identification** that shall be used to reference the view exchange protocol defined in this part of ISO 13584.

The **ISO13584_101_vcv_range_constraints_schema** contains two rules, the role of which being to specify some constraints on any **abstract_functional_model_class** that references the view exchange protocol defined in this part of ISO 13584.

NOTE 2   **abstract_functional_model_class** is defined in ISO 13584.

All these rules shall be fulfilled by any library delivery file that references the view exchange protocol defined in this part of ISO 13584 in any of its conformance class.

When the view exchange protocol defined in this part of ISO 13584 is referenced by a library delivery file, these rules shall be added to the EXPRESS schema specifying the requirements of the library integrated information model referenced by this library delivery file using the following process.

Assume that a library delivery file references a library integrated information model "L", the requirements of which are stated in the unique schema without external references called "L_library_implicit_schema", and the view exchange protocol defined in this part of ISO 13584.

NOTE 3   An EXPRESS schema that specifies a set of requirements without any external references is often called a long form schema.

EXAMPLE 1   **ISO13584_f_m_iim_library_implicit_schema** is an EXPRESS schema that specifies the requirement of LIIM 24-2 by means of a single schema without any external references.

NOTE 4   **ISO13584_f_m_iim_library_implicit_schema** is defined in annex G of ISO 13584-24. This library delivery file shall fulfill the requirements of the schema defined as follows:

— Check that all the entities referenced in the **ISO13584_101_cc_1_or_2_or_3_schema**, **ISO13584_101_cc_1E_or_2E_or_3E_schema** and **ISO13584_101_vcv_range_constraints_schema** schemas are already existing in the **L_library_implicit_schema**, else reference to the library integrated information model "L" and to the view exchange protocol defined in this part of ISO 13584 is not allowed.

NOTE 5   The information model of a library delivery file and the entities it may contain are specified by a library integrated information model. A view exchange protocol may only add constraints.

EXAMPLE 2   The view exchange protocol defined in this part of ISO 13584 references the **abstract_functional_model_class** entity. It cannot be used with the **ISO13584_g_m_iim_library_implicit_schema** that specifies the requirement of LIIM 24-1 and does not reference any EXPRESS resource construct for modeling functional models.

NOTE 6   **ISO13584_g_m_iim_library_implicit_schema** is defined in annex C of ISO 13584-24.

— Replace in the **ISO13584_101_cc_1_or_2_or_3_schema** schema, and in the **ISO13584_101_cc_1E_or_2E_or_3E_schema** schema the string 'ISO13584_EXTERNAL_FILE_SCHEMA' by the string 'L_library_implicit_schema' in capital letters.

EXAMPLE 3   If the library integrated information model "L" is the LIIM 24-2 defined in ISO 13584-24, replace the string 'ISO13584_EXTERNAL_FILE_SCHEMA' by the string 'ISO13584_F_M_IIM_LIBRARY_IMPLICIT_SCHEMA'.

— Add to the **L_library_implicit_schema** all the functions and rules defined in the **ISO13584_101_cc_1_or_2_or_3_schema**, **ISO13584_101_cc_1E_or_2E_or_3E_schema** and **ISO13584_101_vcv_range_constraints_schema** schemas.

NOTE 7   The result of the above process is that the **ISO13584_101_side_and_geometry_level_compatibility_rule**, **ISO13584_101_variant_and_unregistered_variant_compatibility_rule,** **ISO13584_101_allowed_reference_to_conformance_class_1_2_and_3_rule** and

ISO13584_101_allowed_reference_to_conformance_class_1E_2E_and_3E_rule rules have been added in the **L_library_implicit_schema** together with all the functions required to ensure these rules.

NOTE 8   The names of the functions and rules defined in this part of ISO 13584 are prefixed by the string 'ISO13584_101' to avoid name conflicts when several view exchange protocols are referenced by the same library delivery file.

NOTE 9   A library delivery file referencing a library integrated information model "L", the requirements of which are stated in the unique schema without external references called "L_library_implicit_schema", and the view exchange protocol defined in this part of ISO 13584 may also be exchanged using the **L_library_implicit_schema** defined in the part of ISO 13584 that specifies the library integrated information model "L" without any further change. In this case the requirements specified in this part of ISO 13584 are not included in the exchange schema. They shall be checked before and after exchange of the library delivery file.

This listing of the **ISO13584_101_cc_1_or_2_or_3_schema**, **ISO13584_101_cc_1E_or_2E_or_3E_schema** and **ISO13584_101_vcv_range_constraints_schema** schemas are available in computer-interpretable form and can be found at the following URL:

> http://www.tc184-sc4.org/EXPRESS/

If there is difficulty accessing these sites, contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE 10   The information provided in computer-interpretable form at the above URLs is normative.

NOTE 11   If some errors are identified in the EXPRESS code during the ballot process, the description of these errors, together with the corrections recommended for PLIB implementations by the part editors can be found at the following URL:

> http://www.lisi.ensma.fr/ftp/pub/PLIB_release_notes/Part101/Part101-IS/

6.3.1 Conformance class specification table

Table 4 specifies the values of **view_exchange_protocol_identification.name** and **view_exchange_protocol_identification.application** that are allowed for use in a **view_exchange_protocol_identification** to reference the view exchange protocol defined in this part of ISO 13584 in either of its conformance classes.

Table 4 — ISO 13584-101 conformance class specification

Conformance Class	view_exchange_protocol_identification .name mandatory value	view_exchange_protocol_identification .application mandatory value
1	'ISO_13584_101'	'1'
2	'ISO_13584_101'	'2'
3	'ISO_13584_101'	'3'
1E	'ISO_13584_101'	'1E'
2E	'ISO_13584_101'	'2E'
3E	'ISO_13584_101'	'3E'

6.3.2 Constraints on a library delivery file referencing *basic_geometry*

The **view_control_variable_range** instance values allowed for use in a library delivery file to reference the *basic_geometry* functional view class defined in this part of ISO 13584 in any of its conformance classes shall obey the constraints defined in the following EXPRESS schema.

EXPRESS specification:

```
*)
SCHEMA ISO13584_101_vcv_range_constraints_schema;

REFERENCE FROM ISO13584_extended_dictionary_schema(
    abstract_functional_model_class,
    view_control_variable_range);
(*
```

NOTE   The schema used above can be found in the following document:
ISO13584_extended_dictionary_schema   ISO 13584-24.

6.3.2.1 ISO13584_101_side_and_geometry_level_compatibility_rule rule

The **ISO13584_101_side_and_geometry_level_compatibility_rule** stipulates that the **abstract_functional_model_class**es that reference the *basic_geometry* representation category by its **created_view** attribute shall always reference in its **v_c_v_range**:

— in case of a two-dimensional geometric representation, a range that contains or not the null value for side view control variable, and

   NOTE   A two-dimensional isometric view is an example of a view of which the side view control variable should take the null value.

— in case of a three-dimensional geometric representation, a range that consists only of the null value for the side view control variable, and

— in case of a two-dimensional or three-dimensional, provide the allowed side values together with a null value for the side view control variable.

EXAMPLE The following geometry level and side **view_control_variable_range**s are allowed:

Geometry level **view_control_variable_range**	Side **view_control_variable_range**
[1:1]	[0:6]
[1:1]	[1:6]
[1:3]	[0:6]
[2:2]	[0:0]

The following geometry level and side **view_control_variable_range**s are not allowed:

Geometry level **view_control_variable_range**	Side **view_control_variable_range**
[1:2]	[1:6]
[2:2]	[1:6]

<u>EXPRESS specification:</u>

```
*)
RULE ISO13584_101_side_and_geometry_level_compatibility_rule FOR
      (abstract_functional_model_class);
LOCAL
      geometry_level_set: SET OF view_control_variable_range;
      side_set: SET OF view_control_variable_range;
      is_2D, is_2D_or_3D, is_3D: BOOLEAN;
      no_side, with_side, side_or_no_side, res: BOOLEAN;
      compatible_side_and_geometry_level: BOOLEAN := TRUE;
END_LOCAL;

REPEAT i := 1 TO SIZEOF(abstract_functional_model_class);

      geometry_level_set := QUERY(vcvr <*
          abstract_functional_model_class[i].v_c_v_range |
          (vcvr.parameter_type.name_scope.code = 'basic_geometry')
          AND (vcvr.parameter_type.name_scope.defined_by.code =
              '0112/1///13584_101_1')
          AND(vcvr.parameter_type.code = 'geometry_level'));

      side_set := QUERY(vcvr <*
          abstract_functional_model_class[i].v_c_v_range |
          (vcvr.parameter_type.name_scope.code = 'basic_geometry')
          AND (vcvr.parameter_type.name_scope.defined_by.code =
              '0112/1///13584_101_1')
          AND (vcvr.parameter_type.code = 'side'));

      is_2D := (SIZEOF(QUERY(vcvr <* geometry_level_set |
          ((vcvr.range_lobound = 1) AND (vcvr.range_hibound = 1))))
          = 1);
      is_2D_or_3D := (SIZEOF(QUERY(vcvr <* geometry_level_set |
          ((vcvr.range_lobound = 1) AND (vcvr.range_hibound > 1))))
          = 1);
      is_3D := (SIZEOF(QUERY(vcvr <* geometry_level_set |
          (vcvr.range_lobound > 1) AND (vcvr.range_hibound > 1))) = 1);

      no_side := (SIZEOF(QUERY(vcvr <* side_set |
          ((vcvr.range_lobound = 0) AND (vcvr.range_hibound = 0))))
          = 1);
      side_or_no_side := (SIZEOF(QUERY(vcvr <* side_set |
          ((vcvr.range_lobound = 0) AND (vcvr.range_hibound >= 0))))
          = 1);

      IF is_2D OR ((NOT(is_2D_or_3D) OR
          side_or_no_side) AND (NOT(is_3D) OR no_side))
      THEN
          res := TRUE;
      ELSE
          res := FALSE;
      END_IF;
```

```
        compatible_side_and_geometry_level :=
            compatible_side_and_geometry_level AND res;
    END_REPEAT;

    WHERE
        WR1: compatible_side_and_geometry_level;
    END_RULE; -- ISO13584_101_side_and_geometry_level_compatibility_rule
    (*
```

Formal propositions:

WR1: each **abstract_functional_model_class** that references a *basic_geometry* representation category shall define a **view_control_variable** specifying the geometric representation side for which **range_lobound** and **range_hibound** are equal to 0 in case of a three-dimensional representation, or for which **range_lobound** is equal to 0 and **range_hibound** is greater or equal to 0 in case of a two-dimensional and three-dimensional geometric representations.

Informal propositions:

IP1: no method of any **abstract_functional_model_class** included in the library delivery file shall create a view whose side view control variable is different from 'null' and whose geometry level view control variable is different from '2D'.

6.3.2.2 ISO13584_101_variant_and_unregistered_variant_compatibility_rule rule

The **ISO13584_101_ISO13584_101_variant_and_unregistered_variant_compatibility_rule** rule stipulates the **abstract_functional_model_class**es that provide the *basic_geometry* representation category in different *variant*s or *unreg_variant*s to always provide:

— a standardised shape representation and/or a library data supplier dependent shape representation, and

— in case of a possible standardized shape representation, at the minimum a *variant* '1', i.e., 'external shape', and

— in case of a possible library data supplier dependent shape representation, an *unreg_variant* strictly greater than 0.

EXAMPLE The following variant and unregistered variant **view_control_variable_range**s are allowed:

Variant **view_control_variable_range**	Unregistered variant **view_control_variable_range**
[0:0]	[3:5]
[1:3]	[0:0]
[0:2]	[0:1]

The following variant and unregistered variant **view_control_variable_range**s are not allowed:

Variant **view_control_variable_range**	Unregistered variant **view_control_variable_range**
[0:0]	[0:0]
[0:0]	[0:2]

EXPRESS specification:

```
    *)
    RULE   ISO13584_101_variant_and_unregistered_variant_compatibility_rule
    FOR
        (abstract_functional_model_class);
    LOCAL
        variant_set: SET OF view_control_variable_range := [];
        unreg_variant_set: SET OF view_control_variable_range := [];
        is_variant, is_unreg_variant: BOOLEAN;
        may_be_variant, may_be_unreg_variant: BOOLEAN;
        variant_1_mandatory, unreg_variant_greater_than_0: BOOLEAN;
        res, compatible_variant_and_unreg_variant: BOOLEAN := TRUE;
    END_LOCAL;

    REPEAT i := 1 TO SIZEOF(abstract_functional_model_class);

        variant_set := QUERY(vcvr <*
            abstract_functional_model_class[i].v_c_v_range |
                (vcvr.parameter_type.name_scope.code = 'basic_geometry')
            AND (vcvr.parameter_type.name_scope.defined_by.code =
                '0112/1///13584_101_1')
            AND(vcvr.parameter_type.code = 'variant'));

        unreg_variant_set := QUERY(vcvr <*
            abstract_functional_model_class[i].v_c_v_range |
                (vcvr.parameter_type.name_scope.code = 'basic_geometry')
            AND (vcvr.parameter_type.name_scope.defined_by.code =
                '0112/1///13584_101_1')
            AND (vcvr.parameter_type.code = 'unreg_variant'));

        is_unreg_variant := (SIZEOF(QUERY(vcvr <* variant_set
            |(vcvr.range_lobound = 0) AND (vcvr.range_hibound = 0)))
            = 1);
        may_be_unreg_variant := (SIZEOF(QUERY(vcvr <* variant_set
            |(vcvr.range_lobound = 0) AND (vcvr.range_hibound > 0)))
            = 1);
        variant_1_mandatory := (SIZEOF(QUERY(vcvr <* variant_set
            |(vcvr.range_lobound = 1))) = 1);
        is_variant := (SIZEOF(QUERY(vcvr <* unreg_variant_set
            |(vcvr.range_lobound = 0) AND (vcvr.range_hibound = 0)))
            = 1);
        may_be_variant := (SIZEOF(QUERY(vcvr <* unreg_variant_set
            |(vcvr.range_lobound = 0) AND (vcvr.range_hibound > 0)))
            = 1);
        unreg_variant_greater_than_0 := (SIZEOF(QUERY(vcvr <*
            unreg_variant_set |(vcvr.range_lobound >= 1))) = 1);

        IF (is_variant AND variant_1_mandatory) OR (is_unreg_variant AND
            unreg_variant_greater_than_0) OR (may_be_unreg_variant AND
            may_be_variant)
        THEN
            res := TRUE;
```

```
            ELSE
                res := FALSE;
            END_IF;

            compatible_variant_and_unreg_variant :=
                compatible_variant_and_unreg_variant AND res;
            variant_set := [];
            unreg_variant_set := [];
    END_REPEAT;

    WHERE
        WR1: compatible_variant_and_unreg_variant;
    END_RULE;                                                   --
    ISO13584_101_variant_and_unregistered_variant_compatibility_rule
    (*
```

Formal propositions:

WR1: each **abstract_functional_model_class** that references a *basic_geometry* representation
category shall define two **view_control_variable**s defining respectively a variant and an unregistered
variant, for which values are compatible: either the variant **view_control_variable_range** range is
limited to 0 and the unregistered variant **view_control_variable_range** range is any range but the 0
value, or the unregistered variant **view_control_variable_range** range is 0 and the variant
view_control_variable_range range is from 1 to any value, or both variant
view_control_variable_range and unregistered variant **view_control_variable_range** ranges are
specified in a range from 0 to any value greater or equal than 1.

```
    *)

    END_SCHEMA; -- ISO13584_101_vcv_range_constraints_schema

    (*
```

6.3.3 Constraints on a library delivery file for referencing conformance class 1, 2 and 3

The **view_exchange_protocol_identification** instance values allowed for use in a library delivery file
to reference the view exchange protocol defined in this part of ISO 13584 in any of its conformance
classes 1, 2 or 3 shall obey the constraints defined in the following EXPRESS schema.

EXPRESS specification:

```
    *)
    SCHEMA ISO13584_101_cc_1_or_2_or_3_schema;

    REFERENCE FROM ISO13584_IEC61360_dictionary_schema(
        item_names);

    REFERENCE FROM ISO13584_extended_dictionary_schema(
        data_exchange_specification_identification,
        view_exchange_protocol_identification);

    REFERENCE FROM ISO13584_external_file_schema(
        external_file_protocol,
        program_status,
```

```
      simple_program_protocol,
      standard_simple_program_protocol);


   REFERENCE FROM person_organization_schema(
      organization);
(*
```

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

ISO13584_IEC61360_dictionary_schema	Informative annex of ISO 13584-42 and normative annex in IEC 61360-2,
ISO13584_extended_dictionary_schema	ISO 13584-24,
ISO13584_external_file_schema	ISO 13584-24,
person_organization_schema	ISO 10303-41.

6.3.3.1 ISO13584_101_allowed_reference_to_conformance_class_1_2_and_3_rule rule

The **ISO13584_101_allowed_reference_to_conformance_class_1_2_and_3_rule** rule defines constraints on a **view_exchange_protocol_identification**s to be allowed for use to reference conformance class 1, 2 or 3 of the view exchange protocol defined in this part of ISO 13584. A **view_exchange_protocol_identification** is allowed for use to reference conformance class 1, 2 or 3 of the view exchange protocol defined in this part of ISO 13584 if the following conditions hold:

— the **name** attribute of the **view_exchange_protocol_identification** is equal to 'ISO_13584_101', and

— the **external_file_protocols** attribute of the **view_exchange_protocol_identification** references only one **external_file_protocol**, and

— the **level** attribute of the referenced **external_file_protocol** shall be equal to the **application** attribute of the **view_exchange_protocol_identification**, and

— the **status** attribute of the **view_exchange_protocol_identification** shall be equal to either 'WD', 'CD', 'DIS', 'FDIS' or 'IS', and

— the **application** attribute of the **view_exchange_protocol_identification** shall have '1', '2' or '3' as its value, and

— the **external_file_protocol** referenced by the **external_file_protocols** attribute of the **view_exchange_protocol_identification** shall fulfill the constraints required by the **ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3** function defined in Clause 6.3.3.2.

<u>EXPRESS specification:</u>

```
   *)
   RULE ISO13584_101_allowed_reference_to_conformance_class_1_2_and_3_rule
   FOR (
      view_exchange_protocol_identification);
   WHERE
   WR1: QUERY(vep_id <* view_exchange_protocol_identification |
      (vep_id\data_exchange_specification_identification.name
      = 'ISO_13584_101')
      AND
      (SIZEOF(vep_id\data_exchange_specification_identification
      .external_file_protocols) <> 1)) = [];
   WR2: QUERY(vep_id <* view_exchange_protocol_identification |
      (vep_id\data_exchange_specification_identification.name
```

```
               = 'ISO_13584_101')
            AND
            (QUERY(efp <* vep_id\data_exchange_specification_identification
            .external_file_protocols | NOT(efp.level = vep_id.application)
            ) <> []))) = [];
   WR3: QUERY(vep_id <* view_exchange_protocol_identification |
            ((vep_id\data_exchange_specification_identification.status
            = 'WD')
            OR (vep_id\data_exchange_specification_identification.status
            = 'CD')
            OR (vep_id\data_exchange_specification_identification.status
            = 'DIS')
            OR (vep_id\data_exchange_specification_identification.status
            = 'FDIS')
            OR (vep_id\data_exchange_specification_identification.status
            = 'IS'))
        AND
            (vep_id\data_exchange_specification_identification.name
            = 'ISO_13584_101')
        AND
            ((vep_id\data_exchange_specification_identification
            .application = '1')
            OR (vep_id\data_exchange_specification_identification
            .application = '2')
            OR (vep_id\data_exchange_specification_identification
            .application = '3'))
        AND
            (QUERY( efp <*
            vep_id\data_exchange_specification_identification
            .external_file_protocols |
            NOT(ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3(efp)))
= []))
        = QUERY(vep_id <* view_exchange_protocol_identification |
            (vep_id\data_exchange_specification_identification.name
            = 'ISO_13584_101')
        AND
            (vep_id\data_exchange_specification_identification
            .application LIKE '#'));
   END_RULE;                                                     --
   ISO13584_101_allowed_reference_to_conformance_class_1_2_and_3_rule
   (*
```

Formal propositions:

WR1: in **view_exchange_protocol_identification**s that reference conformance class 1, 2 or 3 of the view exchange protocol defined in this part of ISO 13584, only one **external_file_protocol** shall be referenced.

WR2: the **level** attribute of the referenced **standard_program_protocol** shall be equal to the **application** attribute of the **view_exchange_protocol_identification**s that reference conformance class 1, 2 or 3 of the view exchange protocol defined in this part of ISO 13584.

WR3: when referencing the view exchange protocol defined in this part of ISO 13584, the **view_exchange_protocol_identification.name** shall have 'ISO_13584_101' as its value,

view_exchange_protocol_identification.status shall be equal to either 'WD', 'CD', 'DIS', 'FDIS' or 'IS', the **view_exchange_protocol_identification.application** shall have '1', '2' or '3' as its value, and the **view_exchange_protocol_identification.external_file_protocols** shall fulfill the constraints required by the **ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3** function defined in Clause 6.3.3.2.

6.3.3.2 ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3 function

The **ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3** function checks whether an **external_file_protocol** may be referenced by **view_exchange_protocol_identification**s that represent conformance class 1, 2 or 3 of this part of ISO 13584 or not. It returns TRUE if the given **external_file_protocol** is allowed for reference, otherwise, it returns FALSE. An **external_file_protocol** may be referenced by a **view_exchange_protocol_identification** that represent conformance class 1, 2 or 3 of this part of ISO 13584 if the following conditions hold:

— the **external_file_protocol** shall be a **standard_simple_program_protocol** of which the **language** attribute is equal to 'FORTRAN', the **status** attribute is equal to SOURCE or COMPILED, and

— the **organization** attribute of the **external_file_protocol** shall reference an **organization** of which the **id** attribute is equal to 'ISO' and the **name** attribute is equal to 'International Organization for Standardization', and

— the **protocol_name** attribute of the **external_file_protocol** shall equal to 'ISO_13584_31', and

— the **level** attribute of the **external_file_protocol**, if it exists, shall have '1', '2' or '3' as its value, and

— the **designation** attribute of the **external_file_protocol** shall reference an **item_names** for which the **preferred_name** attribute is equal to 'ISO_13584_31' and the **short_name** attribute is equal to 'ISO_13584_31'.

EXPRESS specification:

```
    *)
    FUNCTION ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3(
        p: external_file_protocol): BOOLEAN;

        IF (('ISO13584_EXTERNAL_FILE_SCHEMA'
            + '.STANDARD_SIMPLE_PROGRAM_PROTOCOL' IN TYPEOF(p))
            AND
            (ISO13584_101_organization_compliant_to_cc_1_or_2_or_3(
                p.organisation))
            AND (p.protocol_name = 'ISO_13584_31')
            AND (NOT EXISTS(p.level) OR
                ((p.level = '1') OR (p.level = '2') OR (p.level = '3')))
            AND (ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3(
                p.designation))
            AND (p\simple_program_protocol.language = 'FORTRAN')
            AND ((p\simple_program_protocol.status = SOURCE)
                OR (p\simple_program_protocol.status = COMPILED)))
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
        END_IF;
```

```
        END_FUNCTION; -- ISO13584_101_protocol_compliant_to_cc_1_or_2_or_3
    (*
```

6.3.3.3 ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3 function

The **ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3** function checks whether an **item_names** may be referenced from an **external_file_protocol** referenced by **view_exchange_protocol_identification**s that represents conformance class 1, 2 or 3 of this part of ISO13584 or not. An **item_names** may be referenced from an **external_file_protocol** referenced by **view_exchange_protocol_identification**s that represent conformance class 1, 2 or 3 of this part of ISO13584 if the following conditions hold:

—   the **preferred_name** attribute of the **item_names** is equal to 'ISO_13584_31', and

—   the **short_name** attribute of the **item_names** is equal to 'ISO_13584_31'.

The **ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3** function returns TRUE if the given **item_names** fulfills these constraints otherwise, it returns FALSE.

EXPRESS specification:

```
    *)
    FUNCTION ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3(
            name: item_names): BOOLEAN;
        IF ((name.preferred_name = 'ISO_13584_31')
            AND (name.short_name = 'ISO_13584_31'))
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
        END_IF;
    END_FUNCTION; -- ISO13584_101_item_names_compliant_to_cc_1_or_2_or_3
    (*
```

6.3.3.4 ISO13584_101_organization_compliant_to_cc_1_or_2_or_3 function

The **ISO13584_101_organization_compliant_to_cc_1_or_2_or_3** function checks whether an **organization** may be referenced from an **external_file_protocol** referenced by **view_exchange_protocol_identification**s that represent conformance class 1, 2 or 3 of this part of ISO 13584. An **organization** may be referenced from an **external_file_protocol** referenced by **view_exchange_protocol_identification**s that represent conformance class 1, 2 or 3 of this part of ISO 13584 if the following conditions hold:

—   the **id** attribute of the **organization** is equal to 'ISO', and

—   the **name** attribute of the **organization** is equal to 'International Organization for Standardization'.

The **ISO13584_101_organization_compliant_to_cc_1_or_2_or_3** function returns TRUE if the given **organization** fulfills these constraints otherwise, it returns FALSE.

EXPRESS specification:

```
    *)
    FUNCTION ISO13584_101_organization_compliant_to_cc_1_or_2_or_3(
        org: organization): BOOLEAN;
```

```
        IF ((org.id = 'ISO') AND (org.name =
            'International Organization for Standardization'))
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
        END_IF;
    END_FUNCTION; -- ISO13584_101_organization_compliant_to_cc_1_or_2_or_3


    END_SCHEMA; -- ISO13584_101_cc_1_or_2_or_3_schema

    (*
```

6.3.4 Constraints on a library delivery file for referencing conformance class 1E, 2E and 3E

The **view_exchange_protocol_identification** instance values allowed for use in a library delivery file to reference the view exchange protocol defined in this part of ISO 13584 in any of its conformance classes 1E, 2E or 3E shall obey the constraints defined in the following EXPRESS schema.

<u>EXPRESS specification:</u>

```
    *)
    SCHEMA ISO13584_101_cc_1E_or_2E_or_3E_schema;

    REFERENCE FROM ISO13584_extended_dictionary_schema(
        data_exchange_specification_identification,
        view_exchange_protocol_identification);

    REFERENCE FROM ISO13584_external_file_schema(
        external_file_protocol,
        non_standard_simple_program_protocol);
    (*
```

NOTE   The schemas used above can be found in the following documents:
          ISO13584_extended_dictionary_schema   ISO 13584-24
          ISO13584_external_file_schema   ISO 13584-24

6.3.4.1 ISO13584_101_allowed_reference_to_conformance_class_1E_2E_and_3E_rule rule

The **ISO13584_101_allowed_reference_to_conformance_class_1E_2E_and_3E_rule** rule defines constraints on a **view_exchange_protocol_identification** to be allowed for use to reference conformance class 1E, 2E or 3E of the view exchange protocol defined in this part of ISO 13584. A **view_exchange_protocol_identification** is allowed for use to reference conformance class 1E, 2E or 3E of the view exchange protocol defined in this part of ISO 13584 if the following conditions hold:

— the **name** attribute of the **view_exchange_protocol_identification** is equal to 'ISO_13584_101', and

— the **external_file_protocols** attribute of the **view_exchange_protocol_identification** references only one, **external_file_protocol**, and

— the **status** attribute of the **view_exchange_protocol_identification** shall be equal to either 'WD', 'CD', 'DIS', 'FDIS' or 'IS', and

— the **application** attribute of the **view_exchange_protocol_identification** shall have'1E', '2E' or '3E' as its value, and

— the **external_file_protocol** referenced by the **external_file_protocols** attribute of the **view_exchange_protocol_identification** shall fulfill the constraints required by the **ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E** function defined in Clause 6.3.4.2.

<u>EXPRESS specification:</u>

```
*)
RULE
ISO13584_101_allowed_reference_to_conformance_class_1E_2E_and_3E_rule
FOR (
      view_exchange_protocol_identification);
WHERE
WR1: QUERY(vep_id <* view_exchange_protocol_identification |
      (vep_id\data_exchange_specification_identification.name
      = 'ISO_13584_101')
      AND (SIZEOF(vep_id\data_exchange_specification_identification
      .external_file_protocols) <> 1 )) = [];
WR2: QUERY(vep_id <* view_exchange_protocol_identification |(
            (vep_id\data_exchange_specification_identification.status
            = 'WD')
            OR (vep_id\data_exchange_specification_identification.status
            = 'CD')
            OR (vep_id\data_exchange_specification_identification.status
            = 'DIS')
            OR (vep_id\data_exchange_specification_identification.status
            = 'FDIS')
            OR (vep_id\data_exchange_specification_identification.status
            = 'IS'))
        AND
            (vep_id\data_exchange_specification_identification.name
            = 'ISO_13584_101')
        AND
            ((vep_id\data_exchange_specification_identification
            .application = '1E')
            OR (vep_id\data_exchange_specification_identification
            .application = '2E')
            OR (vep_id\data_exchange_specification_identification
            .application = '3E'))
        AND
            (QUERY(efp <*
            vep_id\data_exchange_specification_identification
            .external_file_protocols
            |
    NOT(ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E(efp))
            ) = []))
        = QUERY(vep_id <* view_exchange_protocol_identification |
            (vep_id\data_exchange_specification_identification.name
            = 'ISO_13584_101')
        AND
```

```
                (vep_id\data_exchange_specification_identification
                .application LIKE '#E'));
    END_RULE;                                             --
    ISO13584_101_allowed_reference_to_conformance_class_1E_2E_and_3E_rule
    (*
```

<u>Formal propositions:</u>

WR1: in **view_exchange_protocol_identification**s that reference conformance class 1E, 2E or 3E of the view exchange protocol defined in this part of ISO 13584, only one **external_file_protocol** shall be referenced.

WR2: when referencing the view exchange protocol defined in this part of ISO 13584, the **view_exchange_protocol_identification.name** shall have 'ISO_13584_101' as its value, **view_exchange_protocol_identification.status** shall be equal to either 'WD', 'CD', 'DIS', 'FDIS' or 'IS', the **view_exchange_protocol_identification.application** shall have '1E', '2E' or '3E' as its value, and the **view_exchange_protocol_identification.external_file_protocols** shall fulfill the constraints required by the **ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E** function defined in Clause 6.3.4.2.

6.3.4.2   ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E function

The **ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E** function checks whether **external_file_protocol** may be referenced by **view_exchange_protocol_identification**s that represent conformance class 1E, 2E or 3E of this part of ISO13584 or not. An **external_file_protocol** may be referenced by a **view_exchange_protocol_identification** that represents conformance class 1E, 2E or 3E of this part of ISO13584 if the following condition holds:

—   the **external_file_protocol** shall be a **non_standard_simple_program_protocol**.

The **ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E** function returns TRUE if the given **external_file_protocol** fulfills this constraint otherwise, it returns FALSE.

<u>EXPRESS specification:</u>

```
    *)
    FUNCTION ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E(
         p: external_file_protocol): BOOLEAN;

         IF (('ISO13584_EXTERNAL_FILE_SCHEMA'
             + '.NON_STANDARD_SIMPLE_PROGRAM_PROTOCOL' IN TYPEOF(p)))
         THEN
             RETURN (TRUE);
         ELSE
             RETURN (FALSE);
         END_IF;
    END_FUNCTION; -- ISO13584_101_protocol_compliant_to_cc_1E_or_2E_or_3E

    END_SCHEMA; -- ISO13584_101_cc_1E_or_2E_or_3E_schema

    (*
```

Annex A
(normative)
Information object registration

A.1 Document identification

In order to provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 13584 part (101) version (1) }

is assigned to this part of ISO 13584. The meaning of this value is defined in ISO 8824-1.

Annex B
(informative)
Physical file example

B.1 Example of physical file using the ISO 13584–101 view exchange protocol

```
*/
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('PLIB FUNCTIONAL MODEL EXAMPLE 1'), '1');
FILE_NAME('P24_fm_implicit_p101.spf',
          '2000-06-05T02:38:14',
          (''),
          ('LISI/ENSMA'),
          'ECCO RUNTIME SYSTEM BUILT-IN PREPROCESSOR V2.3.3',
          'ECCO RUNTIME SYSTEM V2.3.3',
          '');
FILE_SCHEMA(('ISO13584_F_M_IIM_LIBRARY_IMPLICIT_SCHEMA'));
ENDSEC;

DATA;
```

/* Global library description */
```
#2=LIBRARY_IN_STANDARD_FORMAT($, $, $, $, (), #30, #11, (#7), (#12),
(#20, #30, #40), (#50, #60, #140, #130), (), #3, $, $, ());
#3=ITEM_NAMES(LABEL('Functional geometry model'), (), LABEL(''), $,
$);
#6=ORGANIZATION('ISO',          'International        Organization        for
Standardization', '');
#7=STANDARD_SIMPLE_PROGRAM_PROTOCOL(#6, $, 'ISO_13584_31', '001', '1',
#8, $, 'FORTRAN', .SOURCE., $, $, $);
#8=ITEM_NAMES(LABEL('ISO_13584_31'), (), LABEL('ISO_13584_31'), $, $);
#11=LIBRARY_IIM_IDENTIFICATION($, 'IS', 'ISO_13584_24_2', 2001, '3', $,
());
#12=VIEW_EXCHANGE_PROTOCOL_IDENTIFICATION($,   'IS',   'ISO_13584_101',
2001, '1', '1', (#7), $);
#10=GLOBAL_LANGUAGE_ASSIGNMENT('en');
```

/* DICTIONARY DESCRIPTION */
```
/* BSU for supplier */
#20=SUPPLIER_BSU('INA', *);  /* INA code unknown */
#30=SUPPLIER_BSU('9/19860073600021', *);
   /* LISI/ENSMA code in the coding scheme ICD=0009 : SIRET number */
#40=SUPPLIER_BSU('0112/1///13584_101_1', *);
   /* Identification of ISO 13584-101 according to ISO 13584-26 */
```

/* BSU for component_class */
```
#50=CLASS_BSU('Bearing', '001', #20);
#60=CLASS_BSU('PAW', '001', #20);
#130=CLASS_BSU('PAW_Geometry', '001', #30);
#140=CLASS_BSU('basic_geometry', '001', #40);
```

/* BSU for properties */
```
#90=PROPERTY_BSU('d_in', '001', #50);
#100=PROPERTY_BSU('d_out', '001', #50);
#110=PROPERTY_BSU('e', '001', #50);
#150=PROPERTY_BSU('geometry_level', '001', #140);
#160=PROPERTY_BSU('detail_level', '001', #140);
#170=PROPERTY_BSU('side', '001', #140);
#180=PROPERTY_BSU('prg', '001', #130);
#190=PROPERTY_BSU('required_side', '001', #130);
#200=PROPERTY_BSU('variant', '001', #140);
#210=PROPERTY_BSU('unreg_variant', '001', #140);
```

/* BSU for table */
```
#230=TABLE_BSU('T2', '001', #130);
```

/* v_c_v range */
```
#155=VIEW_CONTROL_VARIABLE_RANGE(#150, 1, 1);
#165=VIEW_CONTROL_VARIABLE_RANGE(#160, 2, 2);
#175=VIEW_CONTROL_VARIABLE_RANGE(#170, 1, 6);
#205=VIEW_CONTROL_VARIABLE_RANGE(#200, 1, 1);
#215=VIEW_CONTROL_VARIABLE_RANGE(#210, 0, 0);
```

/* supplier description */
```
#31=SUPPLIER_ELEMENT(#30, $, '001', #32, #33);
#32=ORGANIZATION('LISI/ENSMA', 'LISI/ENSMA', '');
#33=ADDRESS($, $, $, $, $, $, $, 'FRANCE', $, $, $, $);
```

/* Dictionary table description */
```
#231=TABLE_ELEMENT(#230, $, '001', #232, TEXT('Definition of the
geometry programs according to the side of the part'), $, $, *, (#196,
#186), (#196));
#232=ITEM_NAMES(LABEL('side / prg table'), (), LABEL(''), $, $);
```

/* Dictionary properties description */
/* prg */
```
#91=REPRESENTATION_P_DET (#180, $, '001', #92, TEXT('variable used to
reference geometry programs'), $, $, $, $, (), $, 'A58', #93, $);
#92=ITEM_NAMES (LABEL('related program'), (), LABEL(''), $, $);
#93=PROGRAM_REFERENCE_TYPE ((
'ISO13584_F_M_IIM_LIBRARY_IMPLICIT_SCHEMA.PROGRAM_REFERENCE'));
```

/* required side */
```
#101=REPRESENTATION_P_DET(#190, $, '001', #102, TEXT('property used
to store the required side'), $, $, $, $, (), $, 'A58', #103, $);
#102=ITEM_NAMES(LABEL('side to be represented'), (), LABEL(''), $,
$);
#103=INT_TYPE('N 1');
```

/* class - table relationship */
```
#1424=CLASS_TABLE_RELATIONSHIP(#71, (#230));
```

/* Dictionary class description */
/* Functional model class view_of definition*/

```
#71=FM_CLASS_VIEW_OF(#130, $, '001', #72, TEXT('Functional model
class describing the 2d standard geometry of PAW'), $, $, $, $, (#180,
#190), (), *, *, *, *, *, #140, (#155, #165, #175, #205, #215), (#150,
#160, #170, #200, #210), (), (), (), (), (), () ,() ,(), #60, (#90,
#100, #110), (),(),());
#72=ITEM_NAMES(LABEL('Functional model class of PAW'), (), LABEL('fm
class of PAW'), $, $);
```

/* Definition of the properties semantics */
```
#176=OPEN_VIEW_PROPERTY_VALUE_SEMANTICS(#170, $);
#186=SELF_PROPERTY_VALUE_SEMANTICS(#180, $);
#196=SELF_PROPERTY_VALUE_SEMANTICS(#190, $);
#206=COLUMN_TRAVERSAL_VARIABLE_SEMANTICS(#2407, #196);

#96 =SELF_PROPERTY_VALUE_SEMANTICS(#90, $);
#106=SELF_PROPERTY_VALUE_SEMANTICS(#100, $);
#116=SELF_PROPERTY_VALUE_SEMANTICS(#110, $);
```

/* Properties syntax definition */
```
#177=INT_NUMERIC_VARIABLE();
#187=ENTITY_INSTANCE_VARIABLE((
'ISO13584_F_M_IIM_LIBRARY_IMPLICIT_SCHEMA.PROGRAM_REFERENCE'));
#197=INT_NUMERIC_VARIABLE();
#207=INT_NUMERIC_VARIABLE();
#97 =REAL_NUMERIC_VARIABLE();
#107=REAL_NUMERIC_VARIABLE();
#117=REAL_NUMERIC_VARIABLE();
```

/* Syntax / semantics association */
```
#178=ENVIRONMENT(#177, #176);
#188=ENVIRONMENT(#187, #186);
#198=ENVIRONMENT(#197, #196);
#208=ENVIRONMENT(#207, #206);
#98 =ENVIRONMENT(#97, #96);
#108=ENVIRONMENT(#107, #106);
#118=ENVIRONMENT(#117, #116);
```

/* LIBRARY DESCRIPTION */

/* Extension of the table */
```
#2300=TABLE_CONTENT(#230, *, (#2301, #2302), '001', '1997-12-19');
#2301=INTEGER_COLUMN((1, 2, 3, 4, 5, 6), 'NR1..1');
#2302=ENTITY_INSTANCE_COLUMN((#2303, #2304, #2305, #2306, #2307,
#2308),
('ISO13584_F_M_IIM_LIBRARY_IMPLICIT_SCHEMA.PROGRAM_REFERENCE'));
#2303=PROGRAM_REFERENCE(#7, #2313, 'Add1_PAW', 'PAW_p1', (), (), ());
#2304=PROGRAM_REFERENCE(#7, #2314, 'Add2_PAW', 'PAW_p2', (), (), ());
#2305=PROGRAM_REFERENCE(#7, #2315, 'Add3_PAW', 'PAW_p3', (), (), ());
#2306=PROGRAM_REFERENCE(#7, #2316, 'Add4_PAW', 'PAW_p4', (), (), ());
#2307=PROGRAM_REFERENCE(#7, #2317, 'Add5_PAW', 'PAW_p5', (), (), ());
#2308=PROGRAM_REFERENCE(#7, #2318, 'Add6_PAW', 'PAW_p6', (), (), ());
#2313=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2323));
#2314=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2324));
```

```
#2315=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2325));
#2316=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2326));
#2317=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2327));
#2318=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2328));
#2323=LANGUAGE_SPECIFIC_CONTENT((#2333), #2333, $);
#2324=LANGUAGE_SPECIFIC_CONTENT((#2334), #2334, $);
#2325=LANGUAGE_SPECIFIC_CONTENT((#2335), #2335, $);
#2326=LANGUAGE_SPECIFIC_CONTENT((#2336), #2336, $);
#2327=LANGUAGE_SPECIFIC_CONTENT((#2337), #2337, $);
#2328=LANGUAGE_SPECIFIC_CONTENT((#2338), #2338, $);
#2333=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2334=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2335=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2336=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2337=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2338=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
```

/* Library definition of the properties */
```
#900 =OPT_OR_MAND_PROPERTY_BSU(#90, .F., .T.);
#1000=OPT_OR_MAND_PROPERTY_BSU(#100, .F., .T.);
#1100=OPT_OR_MAND_PROPERTY_BSU(#110, .F., .T.);
#1700=OPT_OR_MAND_PROPERTY_BSU(#170, .F., .T.);
#1800=OPT_OR_MAND_PROPERTY_BSU(#180, .F., .F.);
#1900=OPT_OR_MAND_PROPERTY_BSU(#190, .F., .F.);
```

/* Functional model class extension */
```
#1300=FUNCTIONAL_MODEL_CLASS_EXTENSION(#130, (#2303, #2304, #2305,
#2306, #2307, #2308), (#7), (#12),'001', '001', (), (), *, *, *, (),
(), (), $, (#900, #1000, #1100), (), (), (#1700, #1800, #1900), (),
(#3000), $, $, (), $);
```

/* Definition of the derivation table that computes 'prg' from 'side' */
```
#2401=FUNCTIONAL_DOMAIN_RESTRICTION((#186), (#176, #196, #206),
(#2402), $);
#2402=GUARDED_FUNCTIONAL_DOMAIN(#2403, #2405);
#2403=OTHERS();
#2405=TABLE_DEFINED_VALUE(#2406);
#2406=SELECT_EXPRESSION((#2407, #2408));
#2407=TABLE_LITERAL(#230);
#2408=EQUALS_EXPRESSION((#207, #177));
```

/* Definition of the methods */
```
#3000=METHOD(#3001, #3002, #7);
#3001=METHOD_SPECIF(#140, (#155, #165, #175, #205, #215), (#90, #100,
#110),
());
#3002=METHOD_BODY((#97, #107, #117, #177, #187, #197, #207),
(#3019));

#3019=METHOD_STATEMENT((#3020));
#3020=GUARDED_STATEMENT(#3021, #3023);
#3021=BOOLEAN_LITERAL(.T.);
#3023=CALL_PROGRAM_STATEMENT(#187, #2401, (#97, #107, #117), (),
```

```
    ()); 

    ENDSEC; 
    END-ISO-10303-21; 

    /*
```

This ISO 10303-21 conformant physical file is associated with six external files that contain the parametric specifications of the PAW geometry for the different 2D side views. This parametric specification is exchanged as parametric programs conformant with ISO 13584-31:1999 and is be generated by some parametric geometry editor.

The following file, named Add1_PAW shows an example of such an automatically generated parametric program. This file has been automatically generated from the EBP (Example Based Programming) system developed in order to support the generation of programs (in this example, the program that generates a front view is called PAW_p1)from a user design.

```
    SUBROUTINE PAW_p1 (d_out, d_int )
    ! implicit declarations
    !
    ! entity types: (d)ir, (p)nt, (l)in, (c)ir,
    !      (g)rp, (s)et, (e)nt, (a)rc or (a)2p
    IMPLICIT INTEGER (d,p,l,c,g,s,e,a, n )
    !
    ! transfo_type(n) are strings that will contain
    ! "mirror", "shift", "rotation" or "homotetia"
    IMPLICIT CHARACTER*(80) (t )
    !
    ! prefix of D.P. var used: (r)adius, (v)al
    IMPLICIT DOUBLE PRECISION (r,v)
    !
    ! global constants
    INTEGER TDB, CAD
    DOUBLE PRECISION ZERO_VALUE
    PARAMETER (TDB = 0,CAD = 1 )
    PARAMETER (ZERO_VALUE = 0.001 )
    INTEGER lstent (1000 )
    INTEGER FALSE, TRUE
    !
    ! parameter declarations:
    !
    DOUBLE PRECISION d_out
    DOUBLE PRECISION d_int
    !
    ! include the types of the P31 and LIB functions
    include 'P31_FUNCTIONS_TYPES'
    include 'LIB_FUNCTIONS_TYPES'
    !
    ! *** Program Body ***
    !
    ! initialise some constant used entities
    grpfix = CREATE_GRP ()
    CALL CLOSE_GRP ()
    a2p_ref = a2p_ref_sys (TDB )
```

```
dir_x = dir_a2p_x (a2p_ref, TDB )
dir_y = dir_a2p_y (a2p_ref, TDB )
pnt_origin = PNT_CARTESIAN_ABSOLUTE(0.0D0, 0.0D0, 0.0D0, TDB )
FALSE = 0
TRUE = 1
CALL INQ_GEOMETRICAL_POWER(POWER,ERR)
three_d = (POWER .GT. 1)
if three_d then
dir_z = dir_a2p_z (a2p_ref, TDB )
end if
!
! *** START OF THE RECORDING SESSION ***
!
! horizontal line of a given y value
pnt1 = PNT_CARTESIAN_ABSOLUTE(0.0D0, 0.0D0, 0.0D0, TDB)
lin1C1 = lib_lin_max(LIN_PNT_LENGTH_DIR(pnt1, 1.0D0, dir_x, TDB ))
CALL ADD_ENT_GRP(grpfix, lin1C1)
!
! vertical line of a given x value
pnt1 = PNT_CARTESIAN_ABSOLUTE(0.0D0, 0.0D0, 0.0D0, TDB)
lin2C1 = lib_lin_max(LIN_PNT_LENGTH_DIR(pnt1, 1.0D0, dir_y, TDB ))
CALL ADD_ENT_GRP(grpfix, lin2C1)
!
! intersection of 2 lines
pntnm1 = PNT_INTERSECTION_2_ENT(lin2C1, lin1C1, TDB)
!
! circle by its centre and its radius
cir3C1 = CIRCLE_RAD_A2P ((d_out) / (2.000D0), &
lib_a2p_pnt(pntnm1 ),false, TDB )
CALL ADD_ENT_GRP(grpfix, cir3C1)
!
! centre of a circle
pntnm2 = PNT_CENTRE_ARC(cir3C1, TDB)
! circle by its centre and its radius
cir4C1 = CIRCLE_RAD_A2P ((d_int) / (2.000D0), &
lib_a2p_pnt(pntnm2 ),false, TDB )
CALL ADD_ENT_GRP(grpfix, cir4C1)
!
! arc by a circle
arc5C1 = cir3C1
CALL ADD_ENT_GRP(grpfix, arc5C1)
!
! arc by a circle
arc6C1 = cir4C1
CALL ADD_ENT_GRP(grpfix, arc6C1)
!
! *** FIX ENTITIES into CAD SYSTEM ***
!
lstent (1 ) = grpfix
CALL FIX_ENT (1,lstent )
RETURN
END
```

Bibliography

[1]   ISO/IEC 1539 (all parts), *Information technology — Programming languages — Fortran*

[2]   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*

Index

ISO 13584-101:2003(E)

ICS 25.040.40

Price based on 40 pages