



DRAFT AMENDMENT ISO 10303-11:1994/DAM 1

ISO/TC 184/SC 4

Secretariat: **ANSI**

Voting begins on
2001-05-17

Voting terminates on
2001-10-17

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Industrial automation systems and integration — Product data representation and exchange —

Part 11:

Description methods: The EXPRESS language reference manual

AMENDMENT 1

iTeh STANDARD PREVIEW

Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits —

(standards.iteh.ai)

Partie 11: Méthodes de description: Manuel de référence du langage EXPRESS

[ISO 10303-11:1994/DAMd 1](#)

AMENDEMENT 1

<https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>

ICS 25.040.40

In accordance with the provisions of Council Resolution 15/1993 this document is circulated in the English language only.

Conformément aux dispositions de la Résolution du Conseil 15/1993, ce document est distribué en version anglaise seulement.

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

Copyright notice

This ISO document is a draft amendment and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to ISO at the address below or ISO's member body in the country of the requester:

Copyright Manager
ISO Central Secretariat
1 rue de Varembe
1211 Geneva 20 Switzerland
tel. + 41 22 749 0111
fax + 41 22 734 0179
e-mail central@iso.ch

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

ITeH STANDARD PREVIEW
(standards.iteh.ai)

[ISO 10303-11:1994/DAmD 1](https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1)

<https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>

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 3.

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.

Amendment 1 to International Standard ISO 10303-11:1994 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

**Industrial automation systems and integration —
Product data representation and exchange —
Part 11: Implementation methods:
The EXPRESS language reference manual**

AMENDMENT 1

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO 10303-11:1994/DAmD 1](https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1)
<https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>

Introduction

This document amends ISO 10303-11:1994/Cor.2:2001; Industrial automation systems and integration — Product data representation and exchange — Part 11: Implementation methods: The EXPRESS language reference manual. The amended document supercedes ISO 10303-11:1994/Cor.2:2001.

The modifications to the text of ISO 10303-11:1994/Cor.2:2001 are of one kind. Each modification is identified as to which kind it represents and is marked with a diamond (◆). The one kind of modification is:

- ◆ (CHANGE): A change to the requirements of ISO 10303-11:1994. Such a modification affects the conformance of implementations.

All changes and additions to all clauses facilitate the development and implementation of the ISO 10303-1000 series of parts called application modules. The scope of this amendment was driven by requirements from the development of the application modules architecture and from the development of application modules themselves. It is expected that the changes and additions will also be beneficial to all users of ISO 10303-11.

Every attempt has been made to keep this amendment from invalidating any EXPRESS schema that conforms to ISO 10303-11:1994/Cor.2:2001. This was a primary requirement on the amendment that limited its scope and technical content. However, several new words have been added to the language: **BASED_ON**, **END_SUBTYPE_CONSTRAINT**, **EXTENSIBLE**, **GENERIC_ENTITY**, **RENAMED**, **SUBTYPE_CONSTRAINT**, **TOTAL_OVER** and **WITH**. Schemas containing these words as EXPRESS identifiers become invalid under this amendment.

NOTE - This amendment is limited in scope and does not eliminate the requirement for the next edition of ISO 10303-11. The technical content of this amendment was written to be compatible with and places constraints on the next edition of ISO 10303-11.

Clause 7.2.1

◆ (CHANGE) Add the new keywords EXTENSIBLE, BASED_ON, WITH, GENERIC_ENTITY, SUBTYPE_CONSTRAINT, END_SUBTYPE_CONSTRAINT, TOTAL_OVER and RENAMED to table 1.

Clause 7.6

◆ (CHANGE) Add a new clause 7.6 after clause 7.5 as follows:

7.6 Language version identification

The **language version** identifier specifies the the edition of the EXPRESS Language Reference Manual upon which a schema is based. This amendment to EXPRESS is version 4. When a schema contains any declaration added to EXPRESS by this amendment, the language version identifier shall be specified.

NOTE - The general form for the value is "v;cc", where v is the version number of this part of ISO 10303, as specified in annex C, and cc is the encoding of conformance class. Future versions of this part of ISO 10303 may specify additional values for v and cc.

EXAMPLE 14a - The following EXPRESS uses the language version identifier:

```
{iso standard 10303 part (11) version (4)}
SCHEMA configuration_controlled_design_of_the_future;
END_SCHEMA;
```

Syntax: <https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ca5943264/iso-10303-11-1994-dam-1>

```
405 language_version_id = ' { iso standard 10303 part (11) version (4) } '.
302 syntax = [ language_version_id ] schema_decl { schema_decl } .
```

Clause 8.4

◆ (CHANGE) Replace the existing 8.4 with the following:

8.4 Constructed data types

There are two kinds of constructed data types in EXPRESS: ENUMERATION data types and SELECT data types. These data types have similar syntactic structures and may only be used to provide underlying representations of defined data types (see 9.1).

8.4.1 Enumeration data type

An ENUMERATION data type has as its domain a set of names. The names are the only valid values of the enumeration data type. An **enumeration_id** is used to designate these names that are referred to as enumeration items. An ENUMERATION data type may be extensible or not.

An ENUMERATION data type that is not extensible and that is not based on an extensible ENUMERATION data type has as its domain the ordered set of enumeration items in its declaration.

An extensible ENUMERATION data type has as its domain the set of the enumeration items in its declaration plus the union of the sets of enumeration items comprising the domains of all

enumeration data types based on the extensible enumeration data type. An enumeration data type that is based on an extensible enumeration data type has as its domain the set of enumeration items in its declaration plus the enumeration items specified directly, not via extension, in the extensible enumeration data type upon which it is based. An enumeration data type may be both an extensible enumeration data type and be based on another extensible enumeration data type. An extensible enumeration data type is a generalisation of the enumeration data types that are based on it. An extensible `ENUMERATION` data type is specified using the `EXTENSIBLE` reserved word. An `ENUMERATION` data type based on an extensible `ENUMERATION` data type is specified using the `BASED_ON` reserved word. An extensible `ENUMERATION` may be specified without enumeration items and may be based on another extensible `ENUMERATION` and not specify any enumeration items extending that base `ENUMERATION`.

An extensible enumeration and its extensions define a domain that consists of `enumeration_ids`. Within that domain all occurrences of the same `enumeration_id` designate the same value even when the `enumeration_id` is specified in multiple enumeration data types contributing to that domain.

NOTE 1 - An extensible enumeration that is extended two or more times in a single context may have a larger domain than its extensions, in which case it really is a generalisation.

EXAMPLE 34 - The following EXPRESS results in a single enumeration item named "red" as both `stop_light` and `canadian_flag` extend the domain of `colour`.

```

TYPE colour = EXTENSIBLE ENUMERATION
END_TYPE;

TYPE stop_light = ENUMERATION BASED_ON colour WITH (red, yellow, green);
END_TYPE;

TYPE canadian_flag = ENUMERATION BASED_ON colour WITH (red, white);
END_TYPE;

```

Syntax:

```

201 enumeration_type = [ EXTENSIBLE ] ENUMERATION [ (( OF
                    enumeration_items ) | enumeration_extension) ] .
402 enumeration_items = '(' enumeration_id { ',' enumeration_id } ')' .
403 enumeration_extension = BASED_ON type_ref [ WITH enumeration_items ] .

```

Two different `ENUMERATION` data types may declare the same `enumeration_id` defined in different domains. In this case, any reference to the `enumeration_id` (e.g. in an expression) shall be pre-qualified with the data type identifier to ensure that the reference is unambiguous. The reference then appears as: `type_id.enumeration_id`.

NOTE 2 - A `type_id` for use in pre-qualifying an `enumeration_id` is always defined as an `ENUMERATION` data type.

NOTE 3 - In the previous edition of this part of ISO 10303 the ordering of the enumeration items defined a value ordering. In this edition of this part of ISO 10303, no ordering is defined except as noted in rule (d) below. This has been changed to allow for extensible enumeration data types, in which the ordering of the extensions cannot be determined.

Rules and Restrictions

- a. An enumeration data type shall only be used as the underlying data type of a defined data type.
- b. Only enumeration data types whose underlying data types are extensible enumerations may be extended.
- c. An enumeration extension may itself be an extensible specification if the reserved word `EXTENSIBLE` is specified in the definition of the enumeration.
- d. For comparison purposes, the ordering of the values of an enumeration that is not extensible and that is not based on an extensible enumeration may be determined by their relative position in the **enumeration_id** list; the first occurring item shall be less than the second; the second less than the third, etc.
- e. There is no ordering of the values of an extensible enumeration or an enumeration that is based on an extensible enumeration.
- f. An enumeration that is not extensible and not based on an extensible enumeration shall specify enumeration items as its domain.
- g. An enumeration that is not extensible and that is based on an extensible enumeration shall specify enumeration items that extend the domain of the extensible enumeration upon which it is based.
- h. The defined data type which declares the enumeration data type shall not include a domain (where) rule.

<https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>
standards.iteh.ai

NOTE 4 - The above rules ensure that a defined data type names an enumeration data type, and the defined data type is not a specialisation of the enumeration data type.

EXAMPLE 34a - This example uses ENUMERATION data types to show how different kinds of vehicles might travel.

```

TYPE car_can_move = ENUMERATION OF (left, right, backward, forward);
END_TYPE;

TYPE plane_can_move = ENUMERATION OF (left, right, backward, forward, up,
down);
END_TYPE;
    
```

The enumeration item `left` has two independent definitions, one being given by each data type of which it is a component. There is no connection between these two definitions of the identifier `left`. A reference to `left` or `right`, by itself, is ambiguous. To resolve the ambiguity, a reference to either of these values must be qualified by the data type name, e.g., `car_can_move.left`.

EXAMPLE 34b - The following example shows how an extensible enumeration may be used to model a context dependent concept of approval. `general_approval`, as its name suggests, is the most general concept of approval explicitly defining only two values. By declaring `general_approval` to be an extensible enumeration allows it to take on context dependent values in schemas which declare extensions to it. If used to represent the domain of attribute then the allowed values of the attribute are context dependent.

```

SCHEMA s1;

TYPE general_approval = EXTENSIBLE ENUMERATION OF (approved, rejected);
    
```

```
END_TYPE;

END_SCHEMA;

SCHEMA s2;

USE FROM s1 (general_approval);

TYPE domain2_approval = EXTENSIBLE ENUMERATION BASED_ON general_approval
WITH (pending);
END_TYPE;

END_SCHEMA;

SCHEMA s3;

USE FROM S1 (general_approval);

TYPE domain3_approval = EXTENSIBLE ENUMERATION BASED_ON general_approval
WITH (cancelled);
END_TYPE;

END_SCHEMA;

SCHEMA s4;

USE FROM s2 (domain2_approval);

REFERENCE FROM S3 (domain3_approval);

TYPE specific_approval = ENUMERATION BASED_ON domain2_approval WITH
(rework);
END_TYPE;

END_SCHEMA;
```

ITeH STANDARD PREVIEW
(standards.iteh.ai)
<https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>
ISO 10303-11:1994/DAmD 1

In the context of schema **S1**:

- **general_approval** has the domain (**approved, rejected**).

In the context of schema **S2**:

- **general_approval** has the domain (**approved, rejected, pending**);
- **domain2_approval** has the domain (**approved, rejected, pending**).

In the context of schema **S3**:

- **general_approval** has the domain (**approved, rejected, cancelled**);
- **domain3_approval** has the domain (**approved, rejected, cancelled**).

In the context of schema **S4**:

- **general_approval** has the domain (**approved, rejected, pending, cancelled, rework**);
- **domain2_approval** has the domain (**approved, rejected, pending, rework**);
- **domain3_approval** has the domain (**approved, rejected, cancelled**);

- **specific_approval** has the domain (**approved, rejected, pending, rework**).

8.4.2 Select data type

A **SELECT** data type defines an underlying representation for a defined type. That underlying representation is a generalization of the named data types in its domain. The defined type for which the **SELECT** data type is the underlying representation may add constraints on its domain by declaring local rules. A **SELECT** data type may be extensible or not.

A **SELECT** data type that is not extensible and that is not based on an extensible **SELECT** data type has as its domain the union of the domains of the named data types in its select list.

An extensible **SELECT** data type has as its domain the union of the domains of the named data types in its select list plus the union of the domains of all select data types based on the extensible select data type. A select data type that is based on an extensible select data type has as its domain the named data types in its select list plus the named data types specified directly, not via extension, in the extensible select data type upon which it is based. A select data type may be both an extensible select data type and be based on another extensible select data type. An extensible **SELECT** data type is specified using the **EXTENSIBLE** reserved word. A **SELECT** data type based on an extensible **SELECT** data type is specified using the **BASED_ON** reserved word. An extensible **SELECT** may be specified without a select list and may be based on another extensible **SELECT** and not specify a select list extending that base **SELECT**.

A **SELECT** data type may be constrained to have only entity instances in its domain by using the reserved word **GENERIC_ENTITY**. In this case all select elements must either be entity data types or select data types whose select list only includes entity data types. If an extensible select data type is constrained to be a generic-entity select data type, then all extensions of that select data type shall be generic-entity select data types, and shall specify the **GENERIC_ENTITY** reserved word.

Syntax:

```

284 select_type = [ EXTENSIBLE ] [ GENERIC_ENTITY ] SELECT
                [ ( select_list | select_extension ) ] .
410 select_list = '(' named_types { ',' named_types } ')'.
409 select_extension = BASED_ON type_ref [ WITH select_list ].
    
```

Rules and Restrictions

- Each item in the select list shall be an entity data type or a defined data type.
- A **SELECT** data type shall only be used as the underlying data type of a defined data type.
- Only select data types based on data types that are extensible selects may be extended.
- A select extension may itself be an extensible specification if the reserved word **EXTENSIBLE** is specified in the definition of the select.
- If the reserved word **GENERIC_ENTITY** is specified then only generic-entity elements are valid in the select list, where generic-entity element is defined as being either an entity data type or a select of generic-entity elements.

- f. A select that is not extensible and not based on an extensible select shall specify a select list as its domain.
- g. A select that is not extensible and that is based on an extensible select shall specify a select list that extends the domain of the extensible select upon which it is based.

NOTE - The value of a SELECT data type may be a value of more than one of the named data types specified in the select list for that select data type.

EXAMPLE 35 - If **a** and **b** are subtypes of **c**, and if they are related by an ANDOR expression, and if there is a data type defined by **SELECT (a,b)**, then it may be that the value of the SELECT data type is an **a** and a **b** at the same time.

EXAMPLE 36 - A choice must be made among several types of things in a given context.

```

TYPE attachment_method = EXTENSIBLE SELECT(nail, screw);
END_TYPE;

TYPE permanent_attachment = SELECT BASED ON attachment_method WITH (glue,
weld);
END_TYPE;

ENTITY nail;
  length : REAL;
  head_area : REAL;
END_ENTITY;

ENTITY screw;
  length : REAL;
  pitch : REAL;
END_ENTITY;

ENTITY glue;
  composition : material_composition;
  solvent : material_composition;
END_ENTITY;

ENTITY weld;
  composition : material_composition;
END_ENTITY;

ENTITY wall_mounting;
  mounting : product;
  on : wall;
  using : attachment_method;
END_ENTITY;

```

A **wall_mounting** attaches a **product** onto a **wall** using an **attachment method**. The initial **attachment method** describes temporary attachment methods. These methods are then extended to add permanent attachment methods. A value of a **wall_mounting** will have a **using** attribute that is a value of one of **nail**, **screw**, **glue** or **weld**.

Clause 8.5

- ◆ (CHANGE) Add generic_entity_type to the end of the list of data types in rule 211 and add the following as the next

to last sentence of paragraph 1:

A GENERIC_ENTITY data type is a generalisation of all entity data types and a subtype of the GENERIC data type.

Clause 8.6

- ◆ (CHANGE) Add a dot in table 7 under the column labeled "a" in the row labeled "Generalized Data Types"

Clause 8.6.1

- ◆ (CHANGE) Replace the second sentence with:

The base data types are the concrete data types and the generalized data types. The concrete data types are the simple data types, the aggregation data types and the named data types. The concrete types are potentially instantiable when used to define the domain of an attribute, the generalized data types are not instantiable when used to define the domain of an attribute.

- ◆ (CHANGE) Add a third paragraph:

The generalized data types shall only be used as base data types in an abstract entity (see 9.7.1.1).

- ◆ (CHANGE) Replace rule 171 with the following rule:

```
171 base_type = concrete_types | generalized_types .
```

Clause 9

iTeh STANDARD PREVIEW

- ◆ (CHANGE) Add "Subtype_constraint" after "Entity;" to the list of principle capabilities in the second paragraph.

[ISO 10303-11:1994/DAmD 1](https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1)

Clause 9.2 <https://standards.iteh.ai/catalog/standards/sist/7bc279d7-6be7-4134-8665-410ea59f4326/iso-10303-11-1994-damd-1>

- ◆ (CHANGE) Replace rule 197 with:

```
197 entity_head = ENTITY entity_id [ABSTRACT] [subsuper] ';' .
```

Clause 9.2.1

- ◆ (CHANGE) Add the following after the second sentence in the first paragraph as part of the first paragraph:

In the case of an abstract entity whose explicit or derived attributes have GENERIC, GENERIC_ENTITY or AGGREGATE as their data type, the attributes must be redeclared with instantiable data types in at least one subtype of the abstract entity in order to create an instance of the abstract entity data type.

Clause 9.2.1.1

- ◆ (CHANGE) Replace rule 167 with:

```
167 attribute_decl = attribute_id | redeclared_attribute .
```

Clause 9.2.1.2

- ◆ (CHANGE) Replace rule 167 with:

```
167 attribute_decl = attribute_id | redeclared_attribute .
```

Clause 9.2.3.4

◆ (CHANGE) Change "five" (ISO 10303-11:Cor.2:2001 changed "three" to "five" and added a new second and third list item) to "following" in the third sentence of the first paragraph and add the following as the last two items in the list after the first paragraph:

- if the original data type of the attributes is a generic-entity data type, then the data type may be changed to a generic-entity select data type;
- an attribute in the supertype may be given a new identifier in the subtype. The new attribute identifier obeys all the scope and visibility rules defined in clause 10.

◆ (CHANGE) Add the following note after the list after the first paragraph:

NOTE - The declaration of a new identifier does not remove the old identifier from the name scope. The old identifier remains available in this entity data type and any subtypes declared for this entity data type.

◆ (CHANGE) Add the following to the Syntax before rule 262.

```
406 redeclared_attribute = qualified_attribute [RENAMED attribute_id ]
```

◆ (CHANGE) Add the following after the last item in the list under Rules and restrictions (ISO 10303-11:Cor.2:2001 added a new (c) and renumbered the previous (c) to (d)):

e) If the attribute is given a new identifier then that identifier shall not be the same as the identifier for any attribute in any supertype of the current entity data type.

◆ (CHANGE) Replace example 49 with the following:

EXAMPLE 49 - Some geometry systems use floating point coordinates while others work in an integer coordinate space.

```
ENTITY point;  
  x : NUMBER;  
  y : NUMBER;  
END_ENTITY;
```

```
ENTITY integer_point  
  SUBTYPE OF (point);  
  SELF\point.x RENAMED integer_x : INTEGER;  
  SELF\point.y RENAMED integer_y : INTEGER;  
END_ENTITY;
```

Clause 9.2.4

◆ (CHANGE) Add the following informative note before the first paragraph:

NOTE - In order that existing schemas remain valid, the declaration of subtype/supertype constraints within the declaration of an entity remains valid under this amendment to ISO 10303-11. However, its use is deprecated and its removal is planned in future editions of ISO 10303-11. The use of the SUBTYPE_CONSTRAINT (see 9.7) in new schema development is encouraged

Clause 9.2.6

◆ (CHANGE) Add the following list item before the last list item in the list after the first paragraph:

- a constructed data type that is based on an extensible constructed data type is a specialization of that extensible constructed data type