INTERNATIONAL STANDARD

ISO 10303-14

First edition
2005-10-01

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

Part 14: Description methods: The EXPRESS-X language reference manual

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

Partie 14: Méthodes descriptives: Le manuel de référence du langage EXPRESS-X

Reference number
ISO 10303-14:2005(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

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 part of ISO 10303 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10303-14 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

This International Standard is organized as a series of parts, each published separately. The structure of this International Standard is described in ISO 10303-1.

Each part of this International Standard is a member of one of the following series: description methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part is a member of the 10 series.

A complete list of parts of ISO 10303 is available from the Internet:

`http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/STEP_(10303)`

Annexes A, B and C form an integral part of this part of ISO 10303. Annexes D, E and F are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. The mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving.

This part of ISO 10303 is a member of the 10 series. This part of ISO 10303 specifies a language for specifying relationships between data that is governed by EXPRESS schemas, and for specifying alternative views of data that is governed by EXPRESS schemas. The language is called EXPRESS-X.

It is assumed that readers of this part of ISO 10303 are familiar with the EXPRESS data specification language defined in ISO 10303-11 and with clear text encoding specification defined in ISO 10303-21.

Industrial automation systems and integration — Product data representation and exchange — Part 14: Description methods: The EXPRESS-X language reference manual

1 Scope

This part of ISO 10303 specifies a language for specifying relationships between data that is governed by EXPRESS schemas, and for specifying alternate views of data that is governed by EXPRESS schemas. The language is called EXPRESS-X.

EXPRESS-X is a structural data mapping language. It consists of language elements that allow an unambiguous specification of a relationship between EXPRESS schemas.

The following are within the scope of this part of ISO 10303:

- mapping of data governed by one EXPRESS schema to data governed by another EXPRESS schema;
- mapping of data governed by one version of an EXPRESS schema to data governed by another version of that EXPRESS schema, where the two schemas have different names;
- specification of requirements for data translators for data sharing and data exchange applications;
- specification of alternate views of data defined by an EXPRESS schema;
- an alternate notation for application protocol mapping tables;
- bi-directional mappings where mathematically possible;
- specification of constraints that may be evaluated against data produced by mapping.

The following are outside the scope of this part of ISO 10303:

- mapping of data defined using means other than EXPRESS;
- identification of the version of an EXPRESS schema;
- graphical representation of constructs in the EXPRESS-X language.

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 8824-1:2002, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1.*

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

ISO 10303-11:2004, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

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

3 Terms and Definitions

3.1 Terms defined in ISO 10303-1

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-1 apply:

– data;

– information.

3.2 Terms defined in ISO 10303-11

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-11 apply:

– complex entity data type;

– complex entity (data type) instance;

– constant;

– entity;

– entity data type;

– entity (data type) instance;

– instance;

– partial complex entity data type;

– partial complex entity value;

– population;

– simple entity (data type) instance;

– subtype/supertype graph;

– token;

– value.

3.3 Other definitions

For the purpose of this part of ISO 10303, the following definitions apply:

3.3.1
binding extent
a set of binding instances constructed from instances in source entity data type extents and view extents

3.3.2
binding instance
a collection of references to entity data type instances and view data type instances associated with a view or map.

3.3.3
entity data type extent
the collection of instances of a given entity data type

3.3.4
EXPRESS-X parser
a tool capable of parsing a specification stated in the EXPRESS-X language

3.3.5
EXPRESS-X mapping engine
a tool that performs structural information mapping based on a specification stated in the EXPRESS-X language

3.3.6
map
the declaration of a relationship between data of one or more source entity data types or source view data types and data of one or more target entity data types

3.3.7
network mapping
a mapping to many target entity instances

3.3.8
qualified binding extent
a subset of a binding extent consisting of binding instances satisfying a set of selection criteria

NOTE A set of selection criteria is satisfied if each selection criterion individually is satisfied.

3.3.9
selection criterion
a logical expression, the criterion being satisfied only if the expression evaluates to the value `TRUE`

3.3.10
source data set
a collection of entity data type instances governed by an EXPRESS schema and serving as an origin of mapping

3.3.11
source extent
a view extent or entity data type extent drawn on to create a binding extent

3.3.12
target data set
a collection of entity instances produced by means of mapping

3.3.13
view
an alternative organization of the information in an EXPRESS schema

3.3.14
view data type
the representation of a view

3.3.15
view data type instance
a named unit of information established by evaluation of a view

3.3.16
view extent
an aggregate of view data type instances that contains all instances that can be constructed from the qualified binding extent

4 Fundamental principles

4.1 Overview

The following principles apply to this part of ISO 10303; the concepts described in ISO 10303-11, clause 5 also apply.

EXPRESS-X provides the specification of:

– differing views of the data governed by an EXPRESS schema, using view declarations (see 9.3) in a schema view (see 9.5);

– the mapping of data governed by one or more source EXPRESS schemas into data governed by one or more target EXPRESS schemas, using map declarations (see 9.4) in a schema map (see 9.6).

An EXPRESS-X schema may contain EXPRESS function and procedure specifications in order to support the definition of views and maps.

NOTE 1 A typographical convention used throughout this specification is to contextualize parts of a sentence to either the discussion of views or maps particularly by including the terms “(views)” or “(maps)” were appropriate in the sentence.

NOTE 2 A typographical convention used throughout this specification is that a binding instance is denoted as an ordered set of entity or view instance names separated by commas "," and enclosed in angle brackets, "<>". The ordering of instance names corresponds to the order of appearance of the source extent in the FROM language element of the subject view or map declaration.

EXAMPLE Given the view declaration:

```
SCHEMA_VIEW my_person_org_schema_view;
REFERENCE FROM person_and_org_schema;
VIEW person_org;
  FROM p: person; o : organization; -- provides ordering
  SELECT
    name : STRING := p.last_name;
    org : STRING := o.department_name;
END_VIEW;
END_SCHEMA_VIEW;
```

the source express schema:

```
SCHEMA person_and_org_schema;
ENTITY person;
  first_name : STRING;
  last_name : STRING;
END_ENTITY;
ENTITY organization;
  department_name : STRING;
END_ENTITY;
END_SCHEMA;
```

and the data (encoded as defined in ISO 10303-21 — see [2]):

```
#1=PERSON('James','Smith');
#2=PERSON('Fredrick','Jones');
#31=ORGANIZATION('Engineering');
#32=ORGANIZATION('Sales');
```


binding instances for this view and data may be written as below. The concept of binding instances is defined in subsequent clauses and is not necessary to understand the example. Note here, however, that the first element of each binding instance is drawn from the **person** extent and the second element is draw from the **organization** extent. This ordering corresponds to the order of appearance of **person** and **organization** in the FROM language element of the view:

```
{<#1,#31>,<#1,#32>,<#2,#31>,<#2,#32>}.
```

4.2 Fundamental principles of the execution model

4.2.1 Overview

This specification defines a language and an execution model. The execution model is composed of two phases: a binding process and an instantiation process. The evaluation of views and maps share a common binding process but differ with respect to instantiation.

4.2.2 Binding process

A binding is an environment in which values are given to variables. A binding instance is a structure that binds the variables declared in the FROM language element of a view or map declaration. The FROM language element references source entity extents and view extents. The values bound are taken from these source extents. Each binding instance is a member of the set computed as the Cartesian product of the source extents referenced. The set of binding instances thus computed is the binding extent of that view or map for the given source extents. The variable bindings of a binding instance provide an environment for the evaluation of the body of the view or map in the instantiation process, where the data referenced in the binding instance is related to structures created in the target population. Thus each

binding instance corresponds to a view data type instance (views) or target entity data type instances (maps) in the target population.

The source extents of maps and views shall be entity data type extents or view extents.

Circularity among references to source extents is prohibited.

EXAMPLE 1 The binding process applied to the view, data and schema defined in the example of 4.1 computes a binding extent of **person_org** {<#1,#31>, <#1,#32>, <#2,#31>, <#2,#32>}. This extent is depicted in tabular form below:

Binding Instance	person			organization	
	#	first_name	last_name	#	department_name
<#1,#31>	#1	'James'	'Smith'	#31	'Engineering'
<#1,#32>	#1	'James'	'Smith'	#32	'Sales'
<#2,#31>	#2	'Fredrick'	'Jones'	#31	'Engineering'
<#2,#32>	#2	'Fredrick'	'Jones'	#32	'Sales'

EXAMPLE 2 The following schema_view is invalid; it contains a cycle of references (view **a** references view **b** which references view **a**).

```
SCHEMA_VIEW invalid;
VIEW a;
  FROM some_b : b;
  attr1 : INTEGER := some_b.attr2 + 2;
END_VIEW;
VIEW b;
  FROM some_a : a;
  attr2 : INTEGER := some_a.attr1 * 3;
END_VIEW;
END_SCHEMA_VIEW;
```


4.2.3 Instantiation process

A binding is an environment in which variables are given values used during the instantiation process. Each binding instance provides a set of values to be bound to the variables. The view instantiation process is the process of evaluating the body of the view (see 9.3.2) for each binding instance in the binding extent. The order of evaluation of the binding instances is not specified.

EXAMPLE 1 The binding process applied to the schema, data, and view declaration of the example in clause 4.1 results in the binding extent of person_org: {<#1,#31>, <#1,#32>, <#2,#31>, <#2,#32>}. The view declaration and data used in that example is repeated here:

```
VIEW person_org;
  FROM p: person; o : organization; -- provides the illustrated ordering
  SELECT
    name : STRING := p.last_name;
    org : STRING := o.department_name;
END_VIEW;
```

```
#1=PERSON('James','Smith');
#2=PERSON('Fredrick','Jones');
#31=ORGANIZATION('Engineering');
#32=ORGANIZATION('Sales');
```


The binding instance <#1,#31> corresponds to the assignment of entity data type instance #1 to the variable p and #31 to variable o. Evaluation of the body of the view in this binding results in a view data type instance with name attribute 'Smith' and org attribute 'Engineering.' View data type instances may be encoded as though they were entity data type instances using the encoding specified in ISO 10303-21 [2]. The view extent for this example is:

```
#100=PERSON_ORG('Smith','Engineering');   /* <#1,#31> */
#101=PERSON_ORG('Smith','Sales');         /* <#1,#32> */
#102=PERSON_ORG('Jones','Engineering');   /* <#2,#31> */
#103=PERSON_ORG('Jones','Sales');         /* <#2,#32> */
```

EXAMPLE 2 A target EXPRESS schema and schema map with structure similar to that of the schema view used in the previous example can be defined as follows:

```
SCHEMA similar_target;
ENTITY person_org;
 name : STRING;
 org : STRING;
END_ENTITY;
END_SCHEMA;
```

```
SCHEMA_MAP similar;
REFERENCE FROM person_and_org_schema AS SOURCE;
REFERENCE FROM similar_target AS TARGET;
MAP person_org_map AS
  po : person_org;
 FROM
  p : person;
  o : organization;
 SELECT
  po.name := p.last_name;
  po.org := o.department_name;
END_MAP;
END_SCHEMA_MAP;
```


Evaluation of the data of the previous example results in the following entity data type instances governed by the schema similar_target:


```
#100=PERSON_ORG('Smith','Engineering');   /* <#1,#31> */
#101=PERSON_ORG('Smith','Sales');         /* <#1,#32> */
#102=PERSON_ORG('Jones','Engineering');   /* <#2,#31> */
#103=PERSON_ORG('Jones','Sales');         /* <#2,#32>  */
```

When the expression in the right-hand-side of an assignment of the body of the map (view) declaration consists only of attribute references (.) from a source entity, and the referenced attribute is an explicit attribute, the mapping of that target entity attribute (view attribute) is bi-directional.

EXAMPLE 3

```
MAP person_org_map AS
 po : person_org;
FROM
 p: person;
 o : organization;
SELECT
 po.name := p.last_name;          -- bi-directional
 po.org := o.department_name;     -- bi-directional
 po.industry_code := o.owning_enterprise.industry.code_num; -- bi-directional
 po.dept_number := dept_func(o.department_name); -- possibly not bi-directional
END_MAP;
```

Whether or not the attribute po.dept_number is bi-directional depends on the nature of the dept_func function.