INTERNATIONAL STANDARD ISO 10303-43:1994
TECHNICAL CORRIGENDUM 1

Published 1999-07-15

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

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

Part 43:
Integrated generic resources: Representation structures

TECHNICAL CORRIGENDUM 1

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

Partie 43: Ressources génériques intégrées: Structures de représentation

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO 10303-43:1994 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

Introduction

This document corrects ISO 10303-43:1994, Product data representation and exchange — Part 43: Integrated generic resources: Representation structures. The corrected document supersedes ISO 10303-43:1994.

The purpose of the modifications to the text of ISO 10303-43:1994 is to correct errors in the EXPRESS definitions likely to cause a compilation problem, to add EXPRESS declarations to correct logical errors in other parts of ISO 10303, to replace the annex for the computer-interpretable EXPRESS with a URL reference, and to replace the object identifier for the document and the schema.

ICS 25.040.40 **Ref. No. ISO 10303-43:1994/Cor.1:1999(E)**

Printed in Switzerland

Modifications to the text of ISO 10303-43:1994

Clause 4, p. 3
The entity measure_value is required to be referenced for the new function valid_measure_value. In the EXPRESS Specification, delete the following:

```
REFERENCE FROM measure_schema
    (measure_with_unit);
```

replace with the following:

```
REFERENCE FROM measure_schema
    (measure_value,
     measure_with_unit);
```

Clause 4.3, p. 8
The EXPRESS type declaration for founded_item_select has to be added to clause 4.3 to correct a logical error in other parts of ISO 10303. The current clause title has to be changed to accomodate the additional type definition. Replace:

4.3 Representation_schema type definition: transformation

with:

4.3 Representation type definitions

4.3.1 founded_item_select

A **founded_item_select** is a selection between a **founded_item** and a **representation_item**.

EXPRESS specification:

```
*)
TYPE founded_item_select = SELECT
  (founded_item,
   representation_item);
END_TYPE;
(*
```

4.3.2 transformation

Clause 4.4.1, p. 8
The EXPRESS specification of uncertainty_measure_with_unit contained logical errors in the WHERE rule. WR1 required modification to the logic and the addition of the function valid_measure_value. Delete the current WR1 and replace WR1 with the following:

```
  WR1: valid_measure_value(SELF\measure_with_unit.value_component);
```

Clause 4.4, p. 17

The EXPRESS entity declaration for founded_item has to be added to clause 4.4 to correct a logical error in other parts of ISO 10303. Add the following new subclause to 4.4 after the current clause 4.4.13.

4.4.14 founded_item

The **founded_item** entity data type represents an element of representation that participates indirectly in the definition of a representation. A **founded_item** can be used only as part of the definition of a **representation_item**, and is founded through the participation of the **representation_item** in a representation. A **founded_item** can not be an item in a **representation**.

NOTE – This entity data type is semantically equivalent to **representation_item**. Its definition as a discrete entity data type allows errors in other parts of ISO 10303 to be corrected in an upwardly compatible manner.

EXPRESS specification:

```
*)
ENTITY founded_item;
END_ENTITY;
(*
```

Informal propositions:

IP1: Each **founded_item** shall participate, directly or indirectly, in the definition of a **representation_item**.

Clause 4.5.3, p. 21
The EXPRESS specification for the function using_representations did not initialize the 'results' variable. In addition, this function required modification to include founded_item in the set of resturned items. Delete clause 4.5.3 and replace with the following:

4.5.3 using_representations

The function **using_representations** returns the set of representations in which a **representation_item** is used.
A **representation_item** is used in a **representation** if it is:

a) referenced in the set of items of the representation,

b) referenced by a **representation_item** used in the **representation**, or

c) referenced by a **founded_item** used in the **representation**.

NOTE – The second and third conditions are checks allowing for a **representation_item** to be used in a **representation** by being part of a tree of related **representation_items** or **founded_-items**. The tree is rooted in an entity used in a **representation** by fulfilling the first condition.

A **founded_item** is used in a **representation** if it is referenced directly, or indirectly, by a **representation_item** in the set of items of the **representation**.

EXPRESS specification:

```
*)
FUNCTION using_representations (item : founded_item_select)
  : SET OF representation;
  LOCAL
    results           : SET OF representation;
    result_bag        : BAG OF representation;
    intermediate_items : SET OF founded_item_select;
  END_LOCAL;
  -- Find the representations in which the item is used and add to the
  -- results set.
  results := [];
  result_bag :=
USEDIN(item,'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag);
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  -- Find all representation_items or founded_items
  -- by which item is referenced directly or indirectly.
  intermediate_items := using_items(item,[]);
  -- If the set of intermediate items is not empty;
  IF SIZEOF(intermediate_items) > 0 THEN
    -- For each element in the set, add the
    -- representations of that element.
    REPEAT i := 1 TO HIINDEX(intermediate_items);
      result_bag := USEDIN(intermediate_items[i],
                    'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS');
      IF SIZEOF(result_bag) > 0 THEN
        REPEAT j := 1 TO HIINDEX(result_bag);
          results := results + result_bag[j];
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_IF;
  -- Return the set of representation in which the input item is
  -- used directly and indirectly (through intervening
  -- representation_items or founded items).
  RETURN (results);
END_FUNCTION;
(*
```


Argument definitions:

item: the **representation_item** or **founded_item** for which using instances of **representation** are determined. This is input to the function.

Clause 4.5, p. 22
The function using_representations requires another function using_items. Add the following new clause 4.5.4 after the current clause 4.5.3.

4.5.4 using_items

The function **using_items** returns the set of instances of **representation_item** or **founded_item** that reference a **representation_item** directly or indirectly.

EXPRESS specification:

```
*)
FUNCTION using_items (item : founded_item_select;
                      checked_items: SET OF founded_item_select)
                    : SET OF founded_item_select;
  LOCAL
    new_check_items    : SET OF founded_item_select;
    result_items       : SET OF founded_item_select;
    next_items         : SET OF founded_item_select;
  END_LOCAL;
  result_items := [];
  new_check_items := checked_items + item;
  -- Find the set of representation_items or founded_items
  -- in which item is used directly.
  next_items := QUERY(z <* bag_to_set( USEDIN(item , '')) |
    ('REPRESENTATION_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) OR
    ('REPRESENTATION_SCHEMA.FOUNDED_ITEM'        IN TYPEOF(z)));
  -- If the set of next_items is not empty;
  IF SIZEOF(next_items) > 0 THEN
    -- For each element in the set, find the using_items recursively
    REPEAT i := 1 TO HIINDEX(next_items);
      -- Check for loop in data model, i.e. one of the next_items
      -- occurred earlier in the set of check_items;
      IF NOT(next_items[i] IN new_check_items) THEN
        result_items := result_items + next_items[i] +
                        using_items(next_items[i],new_check_items);
      END_IF;
    END_REPEAT;
  END_IF;
  -- return the set of representation_items or founded_items
  -- in which the input item is used directly and indirectly.
  RETURN (result_items);
END_FUNCTION;
(*
```


Argument definitions:

item: the **representation_item** for which the referencing instances of **representation_-item** and **founded_item** are determined. This is input to the function.

checked_items: the set of instances of **representation_item** and **founded_item** that have been checked already in order to ensure termination of the recursive function. This is input to the function.

Clause 4.5, p. 22
The function valid_measure_value is required for WR1 of the uncertainty_-measure_with_unit ENTITY definition. Add the following new clause 4.5.5 after the new clause 4.5.4.

4.5.5 valid_measure_value

The function **valid_measure_value** determines whether a **measure_value** is valid. The function returns TRUE if the **measure_value** is numeric and is positive, or if it is textual.

Function **valid_measure_value** returns FALSE otherwise.

EXPRESS specification:

```
*)
FUNCTION valid_measure_value
  (m : measure_value) : BOOLEAN;
  IF ('REAL' IN TYPEOF (m)) THEN
  RETURN (m > 0.0);
  ELSE
    IF ('INTEGER' IN TYPEOF (m)) THEN
    RETURN (m > 0);
    ELSE
      RETURN (TRUE);
    END_IF;
  END_IF;
END_FUNCTION;
(*
```

Argument definitions:

m: the **measure_value** to be checked. This is the input to the function.

Annex A, p. 23
With the addition of the entity founded_item, an additional short name has to be added to Annex A. Add the following row after the FUNCTIONALY_DEFINED_TRANSFORMATION row in Table A.1:

FOUNDED_ITEM	FNDITM

Annex B.1, p. 24
With the changes identified in this Technical Corrigendum, the object identifier for this part of ISO 10303 has changed. Remove the object identifier for the document and replace with the following:

{ iso standard 10303 part(43) version (2) }

Annex B.2, p. 24
With the changes identified in this Technical Corrigendum, the object identifier for the representation-_schema has changed. Remove the object identifier for the representation_schema and replace with the following:

{ iso standard 10303 part(43) version (2) object(1) representation-schema(1) }

Annex C, p. 25
With the changes identified in this Technical Corrigendum, the EXPRESS contained on the diskette is incorrect. Replace the contents of the annex with the following:

This annex provides a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also provides a listing of the complete EXPRESS schema specified in this part of ISO 10303 without comments or other explanatory text. This annex is available in computer-interpretable form and can be found at the following URLs:

Short names: http://www.mel.nist.gov/div826/subject/apde/snr/
EXPRESS: http://www.mel.nist.gov/step/parts/part043/is/tc1/

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 - The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex D, p.26
With the changes identified in this Technical Corrigendium, Figure D.1 has changed. Replace page 26 with the following page: