INTERNATIONAL STANDARD ISO 10303-203:1994
TECHNICAL CORRIGENDUM 2

Published 1998-12-15

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

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

Part 203:
Application protocol: Configuration controlled design

TECHNICAL CORRIGENDUM 2

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

Partie 203: Protocole d'application: Conception contrôlée de configuration

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO 10303-203:1994 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

Introduction

This document corrects ISO 10303-203:1994, Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol: Configuration controlled 3D designs of mechanical parts and assemblies. The corrected document supersedes ISO 10303-203:1994.

The primary purpose of the modifications to the text of ISO 10303-203:1994 is to change the structure and number of conformance classes. This change to the conformance classes documented in clause 6 also affects the Protocol Implementation Conformance Statement (PICS) proforma documented in annex C of the International Standard. This change affects only the requirements to conform to the protocol and does not change the data requirements of the overall International Standard. This change is being performed to facilitate the conformance of mechanical Computer Aided Design (CAD) systems. The original conformance classes still apply; additional, less comprehensive classes have been added.

In addition, several other corrections to the standard are enumerated.

ICS 25.040.40 **Ref. No. ISO 10303-203:1994/Cor.2:1998(E)**

Printed in Switzerland

This page intentionally left blank

Modifications to the text of ISO 10303-203:1994

Foreword, p. viii
Remove the box and enclosed text. Center on the page the following phrase:

This Page Intentionally Left Blank

Clause 3.6, p. 8 and table of contents, p. ii
Remove subclause 3.6.1 and renumber the other subsubclauses accordingly. The term is defined in ISO 10303-42 and is properly referenced in clause 3.3. Remove 3.6.1 and renumber the other subsubclause in the table of contents.

Clause 5.1, table 13, p. 83
There was an erroneous tilde at the end of the 6th line of the reference path for the certification_required attribute of supplied_part_version. Replace the reference path with the following:

product_definition_formation_with_specified_source <=
product_definition_formation <-
product_definition.formation
product_definition <-
product_definition_relationship.related_product_definition
product_definition_relationship =>
supplied_part_relationship
certified_item = supplied_part_relationship
certified_item <-
cc_design_certification.items[i]
cc_design_certification

Clause 5.1, table 13, p. 84
The reference path for the supplier_part_number attribute of supplied_part_version is incorrect. The reference path through supplied_part_relationship is correctly a mapping rule. Replace the reference path with the following:

product_definition_formation_with_specified_source <=
product_definition_formation
{product_definition_formation <-
product_definition.formation
product_definition <-
product_definition_relationship.related_product_definition
product_definition_relationship =>
supplied_part_relationship}
product_definition_formation.of_product ->
product
product.id

Clause 5.1, table 13, p. 85
The first line of the reference path for the assertion supplied_part_version to approval is missing. Replace the reference path with the following:

product_definition_formation_with_specified_source <=
product_definition_formation
{product_definition_formation <-

product_definition.formation
product_definition <-
product_definition_relationship.related_product_definition
product_definition_relationship =>
supplied_part_relationship}
approved_item = product_definition_formation
approved_item <-
cc_design_approval.items[i]
cc_design_approval

Clause 5.1, table 13, p. 86
The reference path for the assertion supplier to supplied_part_version is incomplete. Replace the reference path with the following:

organization <-
person_and_organization.the_organization
person_and_organization <-
person_and_organization_assignment.assigned_person_and_organization
person_and_organization_assignment =>
cc_design_person_and_organization_assignment
cc_design_person_and_organization_assignment.items[i] ->
person_organization_item
person_organization_item = product_definition_formation
product_definition_formation =>
product_definition_formation_with_specified_source

Clause 5.2, p. 91; annex A, p. 289; annex B, p. 403; and annex H, diagram 11, p. 442
Remove the line containing cartesian_transformation_operator_2d from the USE statement for the geometry_schema. Remove the entity cartesian_transformation_operator_2d from annex A. In the entity cartesian_transformation_operator in annex A, replace the supertype of statement with the following:

```
SUPERTYPE OF (cartesian_transformation_operator_3d)
```

In annex B, remove the row containing the entity cartesian_transformation_operator_2d and its corresponding short name from the short name table.

In annex H, remove the entity cartesian_transformation_operator_2d and the lines denoting its relationships to other entities from the diagram. (See change page 442.)

Clause 5.2, p. 91; annex A, p.296; annex B, p. 404; and annex H, diagram 18, p. 449
As written below, add degenerate_toroidal_surface to the USE statement for the geometry_schema between degenerate_pcurve and dimension_count.

```
    degenerate_toroidal_surface,
```

As written below, add the entity degenerate_toroidal_surface to annex A following the entity degenerate_pcurve and preceding the entity design_context.

```
ENTITY degenerate_toroidal_surface
  SUBTYPE OF (toroidal_surface);
    select_outer : BOOLEAN;
  WHERE
```

```
      wr1: (major_radius < minor_radius);
  END_ENTITY; -- degenerate_toroidal_surface
```

In annex B, add a row containing the entity degenerate_toroidal_surface and its corresponding short name 'DGTRSR' to the short names table, as written below.

DEGENERATE_TOROIDAL_SURFACE	DGTRSR

In annex H, add a box labelled 'degenerate_toroidal_surface' connected to toroidal_surface by a heavy line with an open circle at degenerate_toroidal_surface (designating that degenerate_toroidal_surface is a subtype of entity toroidal surface). Also add a fine line from degenerate_toroidal_surface labelled 'select_outer' that terminates in an open circle (designating an attribute select_outer of type BOOLEAN). (See change page 449.)

Clause 5.2, p. 92; annex A, pp. 294 and 311; annex B, p. 406; and annex H, diagram 11, p. 443
Remove the line containing offset_curve_2d from the USE statement for the geometry_schema. Remove the entity offset_curve_2d from annex A. In the entity curve in annex A, replace the supertype of statement with the following:

```
SUPERTYPE OF (ONEOF`(line, conic, pcurve, surface_curve, offset_curve_3d,
     curve_replica))
```

In annex B, remove the row containing the entity offset_curve_2d and its corresponding short name from the short name table.

In annex II, remove the entity offset_curve_2d and the lines denoting its relationships to other entities from diagram 12. (See change page 443.)

Clause 5.2.3.1.4, p. 113 and annex A, p. 304
WR3 of the entity geometrically_bounded_wireframe_shape_representation is incorrect. Replace WR3 with the following:

```
WR3: SIZEOF (QUERY (gcs <* QUERY (it <* SELF\representation.items |
      'CONFIG_CONTROL_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF (it)) |
      NOT (SIZEOF (QUERY (curv <* QUERY (elem <*
      gcs\geometric_set.elements |
      'CONFIG_CONTROL_DESIGN.CURVE' IN TYPEOF (elem)) |
      NOT (valid_geometrically_bounded_wf_curve
      (curv, 'CONFIG_CONTROL_DESIGN')))) = 0))) =0;
```

Clause 5.2.3.2.33, p. 146
In the first paragraph, the reference to ISO 10303-41 is incorrect. Replace the paragraph with the following:

The base definition of the **representation** entity is given in ISO 10303-43. The following modifications apply to this part of ISO 10303.

Clause 5.2.3.2.34, p. 146
In the first paragraph, the reference to ISO 10303-41 is incorrect. Replace the paragraph with the following:

The base definition of the **representation_context** entity is given in ISO 10303-43. The following

modifications apply to this part of ISO 10303.

Clause 5.2.5.12, p. 226 and annex A, p. 373

In the segment of the function gbsf_check_curve restricting polyline to have at least three points, the inequality statement is incorrect and the call to the USEDIN function is also incorrect. Replace the function with the following:

```
FUNCTION gbsf_check_curve (cv : curve; schema_name : STRING) : BOOLEAN;

  -- check whether the input has the right type

  IF NOT (schema_name + '.CURVE' IN TYPEOF (cv)) THEN RETURN(TRUE);
  END_IF;

  -- let those types pass that do not have any further references
  -- respectively rules to be applied

  IF SIZEOF ([schema_name + '.CIRCLE', schema_name + '.ELLIPSE']
    * TYPEOF(cv)) = 1 THEN RETURN(TRUE);
  ELSE

    -- the b_spline_curve shall not self intersect

    IF (schema_name + '.B_SPLINE_CURVE' IN TYPEOF(cv))
      THEN RETURN(NOT (cv\b_spline_curve.self_intersect));
    ELSE

      -- if the curve is a composite_curve, all of its segments shall be valid

      IF schema_name + '.COMPOSITE_CURVE' IN TYPEOF(cv) THEN
        RETURN (SIZEOF (QUERY (seg <* cv\composite_curve.segments |
        NOT (gbsf_check_curve(seg.parent_curve, schema_name)))) = 0);
      ELSE

        -- the curve_replica references other curves that need to be checked

        IF schema_name + '.CURVE_REPLICA' IN TYPEOF(cv) THEN
          RETURN (gbsf_check_curve
          (cv\curve_replica.parent_curve, schema_name));
        ELSE

     -- offset_curve_3d references a curve and shall not self intersect

          IF schema_name + '.OFFSET_CURVE_3D' IN TYPEOF(cv) THEN
            RETURN ((gbsf_check_curve
            (cv\offset_curve_3d.basis_curve, schema_name))
            AND
            (NOT cv\offset_curve_3d.self_intersect));
          ELSE

         -- pcurve references a curve - indirectly, and a basis_surface

            IF schema_name + '.PCURVE' IN TYPEOF(cv) THEN
              RETURN
              ((gbsf_check_curve
```

```
    (cv\pcurve.reference_to_curve\representation.items[1],
    schema_name)) AND
    (gbsf_check_surface
    (cv\pcurve.basis_surface, schema_name)));
  ELSE

    -- polyline shall have at least 3 points and shall only
    -- be used to represent an intersection_curve

    IF schema_name + '.POLYLINE' IN TYPEOF(cv) THEN
      IF (SIZEOF (cv\polyline.points) >= 3)  AND
        (SIZEOF ((bag_to_set (USEDIN (cv,'')) -
        bag_to_set (USEDIN (cv,
        schema_name +
        '.INTERSECTION_CURVE.CURVE_3D')))) = 0)
        THEN RETURN (TRUE);
      END_IF;
    ELSE

      -- surface_curve references a curve_3d and one or two
      -- pcurves or one or two surface_curves or one of each

      IF schema_name + '.SURFACE_CURVE' IN TYPEOF(cv) THEN

        -- if the curve reference is correct, check also the rest

        IF gbsf_check_curve
        (cv\surface_curve.curve_3d, schema_name) THEN
          REPEAT i := 1 TO SIZEOF
          (cv\surface_curve.associated_geometry);

          -- do for one or two associated_geometrys:

          IF schema_name + '.SURFACE' IN TYPEOF
            (cv\surface_curve.associated_geometry[i]) THEN
            IF NOT gbsf_check_surface
              (cv\surface_curve.associated_geometry[i],
              schema_name) THEN
              RETURN(FALSE);
            END_IF;
          ELSE
            IF schema_name + '.PCURVE' IN TYPEOF
              (cv\surface_curve.associated_geometry[i]) THEN
              IF NOT gbsf_check_curve
                (cv\surface_curve.associated_geometry[i],
                schema_name) THEN
                RETURN(FALSE);
              END_IF;
            END_IF;
          END_IF;
        END_REPEAT;
        RETURN(TRUE);
      END_IF;
    ELSE

      -- if the curve is a trimmed_curve
```

```
                IF (schema_name + '.TRIMMED_CURVE') IN TYPEOF (cv) THEN
                -- if a line, parabola or hyperbola is being trimmed,
                -- then valid

                  IF SIZEOF ([schema_name + '.LINE',
                  schema_name + '.PARABOLA',
                  schema_name + '.HYPERBOLA'] *
                  TYPEOF (cv\trimmed_curve.basis_curve)) = 1
                  THEN RETURN (TRUE);

                    -- otherwise, recursively check the basis curve

                    ELSE RETURN (gbsf_check_curve
                      (cv\trimmed_curve.basis_curve, schema_name));
                    END_IF;
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);
END_FUNCTION; -- gbsf_check_curve
```

Clause 5.2.5.13, p. 230 and annex A, p. 377

There is an error in the segment of the function gbsf_check_surface where curves for a simple basis surface are checked. Replace the function with the following:

```
FUNCTION gbsf_check_surface (sf : surface; schema_name : STRING) : BOOLEAN;

  -- check whether the input has the right type

  IF NOT schema_name + '.SURFACE' IN TYPEOF (sf) THEN RETURN(TRUE);
  END_IF;

  -- b_spline_surface has a self_intersect attribute that shall be false

  IF schema_name + '.B_SPLINE_SURFACE' IN TYPEOF(sf) THEN
    RETURN(NOT (sf\b_spline_surface.self_intersect));
  ELSE

    -- basis surface types return true

    IF SIZEOF ([schema_name + '.SPHERICAL_SURFACE',
      schema_name + '.TOROIDAL_SURFACE'] * TYPEOF(sf)) = 3 THEN
      RETURN(TRUE);
    ELSE

      IF schema_name + '.CURVE_BOUNDED_SURFACE' IN TYPEOF(sf) THEN

      -- if there is a simple basis_surface, check the curves
```

```
    IF SIZEOF ([schema_name + '.CONICAL_SURFACE',
      schema_name + '.CYLINDRICAL_SURFACE',
      schema_name + '.PLANE'] * TYPEOF
      (sf\curve_bounded_surface.basis_surface)) = 1 THEN
      RETURN(SIZEOF (QUERY (bcurve <*
      sf\curve_bounded_surface.boundaries |
      NOT (gbsf_check_curve(bcurve, schema_name)))) = 0);
    ELSE

    -- recursively check the basis_surface and then the curves

      IF gbsf_check_surface
      (sf\curve_bounded_surface.basis_surface, schema_name) THEN
        RETURN(SIZEOF (QUERY (bcurve <*
        sf\curve_bounded_surface.boundaries |
        NOT (gbsf_check_curve(bcurve, schema_name)))) = 0);
      END_IF;
    END_IF;
  ELSE

    -- offset_surface references a surface and shall not self_intersect

    IF schema_name + '.OFFSET_SURFACE' IN TYPEOF(sf) THEN
      RETURN ((gbsf_check_surface
      (sf\offset_surface.basis_surface, schema_name))
      AND
      (NOT (sf\offset_surface.self_intersect)));
    ELSE

      -- rectangular_composite_surface references a matrix of surfaces


   IF schema_name + '.RECTANGULAR_COMPOSITE_SURFACE' IN TYPEOF(sf) THEN
        REPEAT i := 1 TO SIZEOF
          (sf\rectangular_composite_surface.segments);
          REPEAT j := 1 TO SIZEOF
            (sf\rectangular_composite_surface.segments[i]);
            IF NOT (gbsf_check_surface
              (sf\rectangular_composite_surface.segments[i][j].
              parent_surface, schema_name))
                THEN RETURN (FALSE);
            END_IF;
          END_REPEAT;
        END_REPEAT;
        RETURN(TRUE);
      ELSE

        -- rectangular_trimmed_surface has a basis_surface

     IF schema_name + '.RECTANGULAR_TRIMMED_SURFACE' IN TYPEOF(sf) THEN

          IF  SIZEOF ([schema_name + '.CONICAL_SURFACE',
            schema_name + '.CYLINDRICAL_SURFACE',
            schema_name + '.PLANE'] * TYPEOF
            (sf\rectangular_trimmed_surface.basis_surface)) = 1 THEN
            RETURN (TRUE);
          ELSE
```

```
                RETURN(gbsf_check_surface
                (sf\rectangular_trimmed_surface.basis_surface,
                schema_name));
              END_IF;
            ELSE

              -- parent_surfaces need to be checked for their validity

              IF schema_name + '.SURFACE_REPLICA' IN TYPEOF (sf) THEN
                RETURN (gbsf_check_surface
                (sf\surface_replica.parent_surface, schema_name));
              ELSE

                -- and the swept_surface must have a valid swept_curve

                IF schema_name + '.SWEPT_SURFACE' IN TYPEOF(sf) THEN
                  RETURN(gbsf_check_curve
                  (sf\swept_surface.swept_curve, schema_name));
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);
END_FUNCTION; -- gbsf_check_surface
```

Annex A, p. 360

The function boolean_choose defined in ISO 10303:42 is missing labels on the generic formal parameters and the return type of the function. Replace the function with the following:

```
FUNCTION boolean_choose
   (b : boolean;
   choice1, choice2 : generic : item) : generic : item;
   IF b THEN
     RETURN (choice1);
   ELSE
     RETURN (choice2);
   END_IF;
END_FUNCTION; -- boolean_choose
```

Annex A, p. 393

The function valid_calendar_date defined in ISO 10303-41 is missing the otherwise statement. Replace the function with the following:

```
FUNCTION valid_calendar_date
  (date : calendar_date) : LOGICAL;
  IF NOT ((1 <= date.day_component) AND (date.day_component <= 31))
  THEN
    RETURN(FALSE);
  END_IF;
  CASE date.month_component OF
    4    : RETURN(( 1<= date.day_component) AND (date.day_component <= 30));
```

```
    6    : RETURN(( 1<= date.day_component) AND (date.day_component <= 30));
    9    : RETURN(( 1<= date.day_component) AND (date.day_component <= 30));
    11   : RETURN(( 1<= date.day_component) AND (date.day_component <= 30));
    2    :
    BEGIN
      IF leap_year(date.year_component)
      THEN
        RETURN(( 1<= date.day_component) AND (date.day_component <= 29));
      ELSE
        RETURN(( 1<= date.day_component) AND (date.day_component <= 28));
      END_IF;
    END;
    OTHERWISE : RETURN(TRUE);
  END_CASE;
END_FUNCTION;  -- valid_calendar_date
```

Annex J, p. 471 and table of contents, p. iv
The computer-interpretable EXPRESS listing has been updated to reflect the changes made to annex A. Both the EXPRESS listing and the short names are being provided as a CRL. A change page is provided for the affected area of the table of contents. Replace annex J with the following:

Annex J
(informative)

Computer interpretable listings

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

EXPRESS: http://www.mel.nist.gov/step/parts/part203/is/tc2/
Short names: http://www.mel.nist.gov/div826/subject/apde/snr/

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.

Clause 6, pp. 243-269; annex C, pp. 412-413; annex E, page 415; and table of contents, p. iii-iv
The purpose of the following modifications to the text of ISO 10303-203:1994 is to change the structure and number of conformance classes. In addition, a small number of typographical and editorial errors has been corrected. The corrections to clause 6 and annex C of ISO 10303-203:1994 have been documented in change pages for readability. The following pages replace clause 6 and annexes C and E of ISO 10303-203:1994 in their entirety. A replacement page for the affected portion of the table of contents is also provided.

Pages 257 to 269 in ISO 10303-203:1994 become blank pages as clause 6 has been shortened.

Figures

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods: ISO 10303-21. Requirements with respect to implementation methods are specified in annex D.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in annex C.

NOTE - ISO 10303-303[1)] defines the abstract test suite to be used in the assessment of conformance. ISO 10103-32 describes the conformance assessment process.

This part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into conformance classes. Twelve conformance classes are defined. Conformance to this part of ISO 10303 requires, as a minimum, conformance to class 1a. Class 1b is a superset of 1a. Options are defined by classes 2 through 6 and may be selected by an implementation in addition to 1a or 1b. Conformance to a particular conformance class requires that all AIM entities, types, and associated constraints defined as part of that class be supported. Support for a particular conformance class requires support of all the options specified in this class.

The conformance classes are characterized as follows:

— Class 1a: Product identification without shape;

— Class 1b: Configuration-controlled design information without shape;

— Class 2a: Class 1a and shapes represented by geometrically bounded wireframe models, surface models, or both;

— Class 2b: Class 1b and shapes represented by geometrically bounded wireframe models, surface models, or both;

— Class 3a: Class 1a and shapes represented by wireframe models with topology;

— Class 3b: Class 1b and shapes represented by wireframe models with topology;

— Class 4a: Class 1a and shapes represented by manifold surface models with topology;

— Class 4b: Class 1b and shapes represented by manifold surface models with topology;

— Class 5a: Class 1a and shapes represented by faceted b-rep;

2) To be published.

— Class 5b: Class 1b and shapes represented by faceted b-rep;

— Class 6a: Class 1a and shapes represented by advanced b-rep;

— Class 6b: Class 1b and shapes represented by advanced b-rep.

Class 1a is a prerequisite for classes 2 through 6. If an implementation conforms to any of classes 2 through 6, then it shall also conform to class 1a.

Conformance classes 2 through 6 are defined in terms of required AIM **shape_representation** subtypes listed in Table 15. Each **shape_representation** subtype specifies in its local domain rules all of the AIM constructs that define the corresponding conformance class.

shape_representation subtype	class 2	class 3	class 4	class 5	class 6
geometrically_bounded_surface_shape_representation	X				
geometrically_bounded_wireframe_shape_representation	X				
edge_based_wireframe_shape_representation		X			
shell_based_wireframe_shape_representation		X			
manifold_surface_shape_representation			X		
faceted_brep_shape_representation				X	
advanced_brep_shape_representation					X

6.1 Conformance class 1a entities

An implementation of conformance class 1a of this part of ISO 10303 shall support the following entities and related constructs:

— application_context;
— application_context_element;
— application_protocol_definition;
— approval;
— approval_assignment;
— approval_date_time;
— approval_person_organization;
— approval_role;
— approval_status;
— calendar_date;
— cc_design_approval;
— cc_design_date_and_time_assignment;
— cc_design_person_and_organization_assignment;
— cc_design_security_classification;

— coordinated_universal_time_offset;
— date;
— date_and_time;
— date_and_time_assignment;
— date_time_role;
— design_context;
— local_time;
— mechanical_context;
— organization;
— person;
— person_and_organization;
— person_and_organization_assignment;
— person_and_organization_role;
— product;
— product_category;
— product_category_relationship;
— product_context;
— product_definition;
— product_definition_context;
— product_definition_formation;
— product_definition_formation_with_specified_source;
— product_definition_shape;
— product_related_product_category;
— property_definition;
— security_classification;
— security_classification_assignment;
— security_classification_level.

6.2 Conformance class 1b entities

An implementation of conformance class 1b of this part of ISO 10303 shall support class 1a plus the following entities and related constructs:

— action;
— action_assignment;
— action_directive;
— action_method;
— action_request_assignment;
— action_request_solution;
— action_request_status;
— action_status;
— address;
— alternate_product_relationship;
— approval_relationship;
— area_measure_with_unit;
— area_unit;
— assembly_component_usage;
— assembly_component_usage_substitute;
— cc_design_certification;
— cc_design_contract;
— cc_design_specification_reference;
— certification;

— certification_assignment;
— certification_type;
— change;
— change_request;
— configuration_design;
— configuration_effectivity;
— configuration_item;
— context_dependent_unit;
— contract;
— contract_assignment;
— contract_type;
— conversion_based_unit;
— dated_effectivity;
— design_make_from_relationship;
— dimensional_exponents;
— directed_action;
— document;
— document_reference;
— document_relationship;
— document_type;
— document_usage_constraint;
— document_with_class;
— effectivity;
— executed_action;
— length_measure_with_unit;
— length_unit;
— lot_effectivity;
— mass_measure_with_unit;
— mass_unit;
— measure_with_unit;
— named_unit;
— next_assembly_usage_occurrence;
— ordinal_date;
— organization_relationship;
— organizational_address;
— organizational_project;
— personal_address;
— product_concept;
— product_concept_context;
— product_definition_effectivity;
— product_definition_relationship;
— product_definition_usage;
— product_definition_with_associated_documents;
— promissory_usage_occurrence;
— quantified_assembly_component_usage;
— serial_numbered_effectivity;
— shape_aspect;
— shape_aspect_relationship;
— si_unit;
— specified_higher_usage_occurrence;
— start_request;

— start_work;
— supplied_part_relationship;
— versioned_action_request;
— volume_measure_with_unit;
— volume_unit;
— week_of_year_and_day_date.

6.3 Conformance class 2 entities

An implementation of conformance class 2 of this part of ISO 10303 shall support class 1a or 1b (designated as 2a, 2b) and the following entities and related constructs:

— axis1_placement;
— axis2_placement_2d;
— axis2_placement_3d;
— b_spline_curve;
— b_spline_curve_with_knots;
— b_spline_surface;
— b_spline_surface_with_knots;
— bezier_curve;
— bezier_surface;
— boundary_curve;
— bounded_curve;
— bounded_surface;
— cartesian_point;
— cartesian_transformation_operator;
— cartesian_transformation_operator_3d;
— circle;
— composite_curve;
— composite_curve_on_surface;
— composite_curve_segment;
— conic;
— conical_surface;
— context_dependent_shape_representation;
— curve;
— curve_bounded_surface;
— curve_replica;
— cylindrical_surface;
— definitional_representation;
— degenerate_pcurve;
— degenerate_toroidal_surface;
— direction;
— elementary_surface;
— ellipse;
— evaluated_degenerate_pcurve;
— functionally_defincd_transformation;
— geometric_curve_set;
— geometric_representation_context;
— geometric_representation_item;
— geometric_set;
— geometrically_bounded_surface_shape_representation;
— geometrically_bounded_wireframe_shape_representation;

— global_uncertainty_assigned_context;
— global_unit_assigned_context;
— hyperbola;
— intersection_curve;
— item_defined_transformation;
— line;
— mapped_item;
— offset_curve_3d;
— offset_surface;
— outer_boundary_curve;
— parabola;
— parametric_representation_context;
— pcurve;
— placement;
— plane;
— plane_angle_measure_with_unit;
— plane_angle_unit;
— point;
— point_on_curve;
— point_on_surface;
— point_replica;
— polyline;
— property_definition_representation;
— quasi_uniform_curve;
— quasi_uniform_surface;
— rational_b_spline_curve;
— rational_b_spline_surface;
— rectangular_composite_surface;
— rectangular_trimmed_surface;
— reparametrised_composite_curve_segment;
— representation;
— representation_context;
— representation_item;
— representation_map;
— representation_relationship;
— representation_relationship_with_transformation;
— seam_curve;
— shape_aspect;
— shape_aspect_relationship;
— shape_definition_representation;
— shape_representation;
— shape_representation_relationship;
— solid_angle_measure_with_unit;
— solid_angle_unit;
— spherical_surface;
— surface;
— surface_curve;
— surface_of_linear_extrusion;
— surface_of_revolution;
— surface_patch;
— surface_replica;

— swept_surface;
— toroidal_surface;
— trimmed_curve;
— uncertainty_measure_with_unit;
— uniform_curve;
— uniform_surface;
— vector.

6.4 Conformance class 3 entities

An implementation of conformance class 3 of this part of ISO 10303 shall support class 1a or 1b (designated as 3a, 3b) and the following entities and related constructs:

— axis2_placement_3d;
— b_spline_curve;
— b_spline_curve_with_knots;
— bezier_curve;
— bounded_curve;
— cartesian_point;
— cartesian_transformation_operator;
— cartesian_transformation_operator_3d;
— circle;
— conic;
— connected_edge_set;
— context_dependent_shape_representation;
— curve;
— curve_replica;
— direction;
— edge;
— edge_based_wireframe_model;
— edge_based_wireframe_shape_representation;
— edge_curve;
— edge_loop;
— ellipse;
— functionally_defined_transformation;
— geometric_representation_context;
— geometric_representation_item;
— global_uncertainty_assigned_context;
— global_unit_assigned_context;
— hyperbola;
— item_defined_transformation;
— line;
— loop;
— mapped_item;
— offset_curve_3d;
— oriented_edge;
— parabola;
— path;
— placement;
— plane_angle_measure_with_unit;
— plane_angle_unit;
— point;

— point_replica;
— polyline;
— property_definition_representation;
— quasi_uniform_curve;
— rational_b_spline_curve;
— representation;
— representation_context;
— representation_item;
— representation_map;
— representation_relationship;
— representation_relationship_with_transformation;
— shape_aspect;
— shape_aspect_relationship;
— shape_definition_representation;
— shape_representation;
— shape_representation_relationship;
— shell_based_wireframe_model;
— shell_based_wireframe_shape_representation;
— solid_angle_measure_with_unit;
— solid_angle_unit;
— topological_representation_item;
— uncertainty_measure_with_unit;
— uniform_curve;
— vector;
— vertex;
— vertex_loop;
— vertex_point;
— vertex_shell;
— wire_shell.

6.5 Conformance class 4 entities

An implementation of conformance class 4 of this part of ISO 10303 shall support class 1a or 1b (designated as 4a, 4b) and the following entities and related constructs:

— advanced_face;
— axis1_placement;
— axis2_placement_2d;
— axis2_placement_3d;
— b_spline_curve;
— b_spline_curve_with_knots;
— b_spline_surface;
— b_spline_surface_with_knots;
— bezier_curve;
— bezier_surface;
— bounded_curve;
— bounded_surface;
— cartesian_point;
— cartesian_transformation_operator;
— cartesian_transformation_operator_3d;
— circle;

— closed_shell;
— composite_curve;
— composite_curve_on_surface;
— composite_curve_segment;
— conic;
— conical_surface;
— connected_face_set;
— context_dependent_shape_representation;
— curve;
— curve_replica;
— cylindrical_surface;
— definitional_representation;
— degenerate_pcurve;
— degenerate_toroidal_surface;
— direction;
— edge;
— edge_curve;
— edge_loop;
— elementary_surface;
— ellipse;
— evaluated_degenerate_pcurve;
— face;
— face_bound;
— face_outer_bound;
— face_surface;
— functionally_defined_transformation;
— geometric_representation_context;
— geometric_representation_item;
— global_uncertainty_assigned_context;
— global_unit_assigned_context;
— hyperbola;
— intersection_curve;
— item_defined_transformation;
— line;
— loop;
— manifold_surface_shape_representation;
— mapped_item;
— offset_curve_3d;
— offset_surface;
— open_shell;
— oriented_closed_shell;
— oriented_edge;
— oriented_face;
— oriented_open_shell;
— oriented_path;
— parabola;
— parametric_representation_context;
— path;
— pcurve;
— placement;
— plane;
— plane_angle_measure_with_unit;

— plane_angle_unit;
— point;
— point_on_curve;
— point_on_surface;
— polyline;
— property_definition_representation;
— quasi_uniform_curve;
— quasi_uniform_surface;
— rational_b_spline_curve;
— rational_b_spline_surface;
— representation;
— representation_context;
— representation_item;
— representation_map;
— representation_relationship;
— representation_relationship_with_transformation;
— seam_curve;
— shape_aspect;
— shape_aspect_relationship;
— shape_definition_representation;
— shape_representation;
— shape_representation_relationship;
— shell_based_surface_model;
— solid_angle_measure_with_unit;
— solid_angle_unit;
— spherical_surface;
— surface;
— surface_curve;
— surface_of_linear_extrusion;
— surface_of_revolution;
— surface_replica;
— swept_surface;
— topological_representation_item;
— toroidal_surface;
— uncertainty_measure_with_unit;
— uniform_curve;
— uniform_surface;
— vector;
— vertex;
— vertex_loop;
— vertex_point.

6.6 Conformance class 5 entities

An implementation of conformance class 5 of this part of ISO 10303 shall support class 1a or 1b (designated as 5a, 5b) and the following entities and related constructs:

— axis2_placement_3d;
— brep_with_voids;
— cartesian_point;
— cartesian_transformation_operator;

— cartesian_transformation_operator_3d;
— closed_shell;
— connected_face_set;
— context_dependent_shape_representation;
— direction;
— edge;
— elementary_surface;
— face;
— face_bound;
— face_outer_bound;
— face_surface;
— faceted_brep;
— faceted_brep_shape_representation;
— functionally_defined_transformation;
— geometric_representation_context;
— geometric_representation_item;
— global_uncertainty_assigned_context;
— global_unit_assigned_context;
— item_defined_transformation;
— loop;
— manifold_solid_brep;
— mapped_item;
— open_shell;
— oriented_closed_shell;
— oriented_edge;
— oriented_face;
— oriented_open_shell;
— oriented_path;
— path;
— placement;
— plane;
— plane_angle_measure_with_unit;
— plane_angle_unit;
— point;
— poly_loop;
— property_definition_representation;
— representation;
— representation_context;
— representation_item;
— representation_map;
— representation_relationship;
— representation_relationship_with_transformation;
— shape_aspect;
— shape_aspect_relationship;
— shape_definition_representation;
— shape_representation;
— shape_representation_relationship;
— solid_angle_measure_with_unit;
— solid_angle_unit;
— solid_model;
— surface;
— topological_representation_item;

— uncertainty_measure_with_unit;
— vector;
— vertex.

6.7 Conformance class 6 entities

An implementation of conformance class 6 of this part of ISO 10303 shall support class 1a or 1b (designated as 6a, 6b) and the following entities and related constructs:

— advanced_brep_shape_representation;
— advanced_face;
— axis1_placement;
— axis2_placement_2d;
— axis2_placement_3d;
— b_spline_curve;
— b_spline_curve_with_knots;
— b_spline_surface;
— b_spline_surface_with_knots;
— bezier_curve;
— bezier_surface;
— bounded_curve;
— bounded_surface;
— brep_with_voids;
— cartesian_point;
— cartesian_transformation_operator;
— cartesian_transformation_operator_3d;
— circle;
— closed_shell;
— composite_curve;
— composite_curve_on_surface;
— composite_curve_segment;
— conic;
— conical_surface;
— connected_face_set;
— context_dependent_shape_representation;
— curve;
— cylindrical_surface;
— definitional_representation;
— degenerate_toroidal_surface;
— direction;
— edge;
— edge_curve;
— edge_loop;
— elementary_surface;
— ellipse;
— face;
— face_bound;
— face_outer_bound;
— face_surface;
— functionally_defined_transformation;
— geometric_representation_context;

— geometric_representation_item;
— global_uncertainty_assigned_context;
— global_unit_assigned_context;
— hyperbola;
— item_defined_transformation;
— line;
— loop;
— manifold_solid_brep;
— mapped_item;
— open_shell;
— oriented_closed_shell;
— oriented_edge;
— oriented_face;
— oriented_open_shell;
— oriented_path;
— parabola;
— parametric_representation_context;
— path;
— pcurve;
— placement;
— plane;
— plane_angle_measure_with_unit;
— plane_angle_unit;
— point;
— polyline;
— property_definition_representation;
— quasi_uniform_curve;
— quasi_uniform_surface;
— rational_b_spline_curve;
— rational_b_spline_surface;
— representation;
— representation_context;
— representation_item;
— representation_map;
— representation_relationship;
— representation_relationship_with_transformation;
— shape_aspect;
— shape_aspect_relationship;
— shape_definition_representation;
— shape_representation;
— shape_representation_relationship;
— solid_angle_measure_with_unit;
— solid_angle_unit;
— solid_model;
— spherical_surface;
— surface;
— surface_curve;
— surface_of_linear_extrusion;
— surface_of_revolution;
— swept_surface;
— topological_representation_item;
— toroidal_surface;

— uncertainty_measure_with_unit;
— uniform_curve;
— uniform_surface;
— vector;
— vertex;
— vertex_loop;
— vertex_point.

This page intentionally left blank

Annex C
(normative)

Protocol Implementation Conformance Statement (PICS) proforma

Overview:

The Protocol Information and Conformance Statement (PICS) proforma supports the conformance assessment of implementations requesting evaluation to this international standard. The PICS proforma contains a series of questions which capture static information about the implementation under test (IUT). This information is used in both statically assessing valid option selection, and in configuring an appropriate conformance test (dynamic assessment) session.

A number of options are identified in this standard for possible use by conforming implementations. Some of these options may be dynamically (run-time) selected for use/non-use, for instance, OPTIONAL attributes of an entity. Others shall be statically (configuration-time) selected for use/non-use, such as a particular style of geometry as defined in a conformance class.

Questions:

For simplicity of reference, an identifier for the product or system with which the tested ISO 10303 implementation is packaged in and/or procured by is required.

1. Product/system identifier (or name): _____

A conforming implementation shall provide at least conformance class 1a functionality for ISO 10303-203. There are twelve classes defined in this international standard. Each class specifies a subset of ISO 10303-203 AIM constructs. These classes are detailed in clause 6 of this document.

2. Claimed classes of conformance (functionality) - circle choices:

   1a-(Product identification without shape)
   1b-(Configuration-controlled design information without shape)
   2a-(Product identification & Surface & Wireframe & No Topology)
   2b-(Configuration-controlled design information & Surface & Wireframe & No Topology)
   3a-(Product identification & Wireframe & Topology)
   3b-(Configuration-controlled design information & Wireframe & Topology)
   4a-(Product identification & Manifold Surfaces & Topology)
   4b-(Configuration-controlled design information & Manifold Surfaces & Topology)
   5a-(Product identification & Faceted BREP)
   5b-(Configuration-controlled design information & Faceted BREP)
   6a-(Product identification & Advanced BREP)
   6b-(Configuration-controlled design information & Advanced BREP)

Conformance to this international standard may be realized in one or more of several different implementation methods. The implementation methods define what types of exchange behavior are required with respect to this International Standard.

3. Claimed implementation forms - circle choices:

exchange structure (ISO 10303-21)

If the implementation receives data that does not comply with the requirements in this International Standard for the selected conformance class(es), or with the requirements of the ISO 10303-20 series of parts for the selected implementation method, it shall execute a default response. A default response shall be statically set.

4. Default Response: _____

A conforming implementation shall maintain the static options selected throughout subsequent dynamic assessment (testing) without requiring modification. In a user environment, a conforming implementation shall permanently maintain the provision of selected static options, or it shall provide users discretionary control over the changing and setting of the static options, or both (depending on the option).

5. Does the IUT provide some user discretion over the changing and setting of static options?

   Yes or No

6. If yes, which ones?

   (a) Conformance class(es): _____

   (b) Default response: _____

A statement of conformance shall include identification of at least one party deeming conformance for the implementation.

7. Evaluator(s) (tester/certifier/accrediter): _____

This page intentionally left blank