

Australian Standard™

**Industrial automation systems and
integration—Physical device control—
Data model for computerized numerical
controllers**

Part 10: General process data



This Australian Standard was prepared by Committee IT-006, Information Technology for Industrial Automation and Integration. It was approved on behalf of the Council of Standards Australia on 30 March 2004 and published on 4 June 2004.

The following are represented on Committee IT-006:

Association of Consulting Engineers Australia
Australian Electrical and Electronic Manufacturers Association
CSIRO Centre for Planning and Design
CSIRO Manufacturing & Infrastructure Technology
Department of Defence (Australia)
Institute of Instrumentation, Control and Automation Australia
Institution of Engineers Australia
Monash University
RMIT University
The University of Melbourne

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at www.standards.com.au and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year, and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at mail@standards.org.au, or write to the Chief Executive, Standards Australia International Ltd, GPO Box 5420, Sydney, NSW 2001.

This Standard was issued in draft form for comment as DR 04103.

Australian Standard™

**Industrial automation systems and
integration—Physical device control—
Data model for computerized numerical
controllers**

Part 10: General process data

First published as AS ISO 14649.10—2004.

COPYRIGHT

© Standards Australia International

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia International Ltd
GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 5926 2

 **Documents**
Document Sharing Hub

PREFACE

This Standard was prepared by the Standards Australia Committee IT-006, Information Technology for Industrial Automation and Integration.

This Standard is identical with, and has been reproduced from, ISO 14649-10:2003, *Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers*, Part 10: *General process data*.

The objective of this Standard is to provide the process data which is generally needed for NC-programming within all machining technologies. These data elements describe the interface between a computerized numerical controller and the programming system.

This Standard is Part 10 of AS ISO 14649—2004, *Industrial automation systems and integration—Physical device control—Data model for computerized numerical controllers*, which is published in parts as follows:

Part 1: Overview and fundamental principles

Part 10: General process data (this Standard)

Part 11: Process data for milling

The terms ‘normative’ and ‘informative’ are used to define the application of the annex to which they apply. A normative annex is an integral part of a standard, whereas an informative annex is only for information and guidance.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this International Standard’ should read ‘this Australian Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>		<i>Australian Standard</i>	
ISO		AS	
286	ISO system of limits and fits	1654	ISO system of limits and fits
286-1	Part 1: Bases of tolerances, deviations and fits	1654.1	Part 1: Bases of tolerances, deviations and fits
286-2	Part 2: Tables of standard tolerance grades and limit deviations for holes and shafts	1654.2	Part 2: Tables of standard tolerance grades and limit deviations for holes and shafts
		AS ISO	
10303	Industrial automation systems and integration—Product data representation and exchange	10303	Industrial automation systems and integration—Product data representation and exchange
10303-1	Part 1: Overview and fundamental principles	10303.1	Part 1: Overview and fundamental principles
10303-11	Part 11: Description methods: The EXPRESS language reference manual	10303.11	Part 11: Description methods: The EXPRESS language reference manual
10303-43	Part 43: Integrated resources: Representation structures	10303.43	Part 43: Integrated resources: Representation structures
10303-203	Part 203: Application protocol: Configuration controlled design	10303.203	Part 203: Application protocol: Configuration controlled design

CONTENTS

Page

Introduction	iv
1 Scope.....	1
2 Normative references	1
3 Terms and definitions.....	2
4 General process data.....	3
4.1 Header and references	3
4.2 General types and definitions.....	4
4.3 Where to start: Project.....	6
4.4 What to machine: Workpiece and manufacturing feature	7
4.5 Catalogue of manufacturing features	9
4.6 To make things happen: Executables.....	50
4.7 How to machine: Operations	66
4.8 To be in full control: Explicit toolpath definition	70
4.9 Rules.....	78
5 Conformance requirements	79
5.1 Conformance class 1 entities	80
5.2 Conformance class 2 entities	80
5.3 Conformance class 3 entities	82
5.4 Conformance class 4 entities	84
5.5 Conformance class 5 entities	86
5.6 Conformance class 6 entities	88
Annex A (normative) EXPRESS expanded listing.....	91
Annex B (normative) Short names of entities	149
Annex C (normative) Implementation method specific requirements	157
Annex D (informative) EXPRESS-G figures	158
Alphabetical index	174

INTRODUCTION

Modern manufacturing enterprises are built from facilities spread around the globe, which contain equipment from hundreds of different manufacturers. Immense volumes of product information must be transferred between the various facilities and machines. Today's digital communications standards have solved the problem of reliably transferring information across global networks. For mechanical parts, the description of product data has been standardized by ISO 10303. This leads to the possibility of using standard data throughout the entire process chain in the manufacturing enterprise. Impediments to realizing this principle are the data formats used at the machine level. Most computer numerical control (CNC) machines are programmed in the ISO 6983 "G and M code" language. Programs are typically generated by computer-aided manufacturing (CAM) systems that use computer-aided design (CAD) information. However, ISO 6983 limits program portability for three reasons. First, the language focuses on programming the tool center path with respect to machine axes, rather than the machining process with respect to the part. Second, the standard defines the syntax of program statements, but in most cases leaves the semantics ambiguous. Third, vendors usually supplement the language with extensions that are not covered in the limited scope of ISO 6983.

ISO 14649 is a new model of data transfer between CAD/CAM systems and CNC machines, which replaces ISO 6983. It remedies the shortcomings of ISO 6983 by specifying machining processes rather than machine tool motion, using the object-oriented concept of Workingsteps. Workingsteps correspond to high-level machining features and associated process parameters. CNCs are responsible for translating Workingsteps to axis motion and tool operation. A major benefit of ISO 14649 is its use of existing data models from ISO 10303. As ISO 14649 provides a comprehensive model of the manufacturing process, it can also be used as the basis for a bi- and multi-directional data exchange between all other information technology systems.

ISO 14649 represents an object oriented, information and context preserving approach for NC-programming, that supersedes data reduction to simple switching instructions or linear and circular movements. As it is object- and feature oriented and describes the machining operations executed on the workpiece, and not machine dependent axis motions, it will be running on different machine tools or controllers. This compatibility will spare all data adaptations by postprocessors, if the new data model is correctly implemented on the NC-controllers. If old NC programs in ISO 6983 are to be used on such controllers, the corresponding interpreters shall be able to process the different NC program types in parallel.

ISO TC 184/SC 1/WG 7 envisions a gradual evolution from ISO 6983 programming to portable feature-based programming. Early adopters of ISO 14649 will certainly support data input of legacy "G and M codes" manually or through programs, just as modern controllers support both command-line interfaces and graphical user interfaces. This will likely be made easier as open-architecture controllers become more prevalent. Therefore, ISO 14649 does not include legacy program statements, which would otherwise dilute the effectiveness of the standard.

AUSTRALIAN STANDARD

Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers —

Part 10: General process data

1 Scope

This part of ISO 14649 specifies the process data which is generally needed for NC-programming within all machining technologies. These data elements describe the interface between a computerized numerical controller and the programming system (i.e. CAM system or shopfloor programming system). On the programming system, the programme for the numerical controller is created. This programme includes geometric and technological information. It can be described using this part of ISO 14649 together with the technology-specific parts (ISO 14649-11, etc.). This part of ISO 14649 provides the control structures for the sequence of programme execution, mainly the sequence of working steps and associated machine functions.

The “machining_schema” defined in this part of ISO 14649 contains the definition of data types which are generally relevant for different technologies (e.g. milling, turning, grinding). The features for non-milling technologies like turning, EDM, etc. will be introduced when the technology specific parts like ISO 14649-12 for turning, ISO 14649-13 for EDM, and ISO 14649-14 for contour cutting of wood and glass are published. It includes the definition of the workpiece, a feature catalogue containing features which might be referenced by several technologies, the general executables and the basis for an operation definition. Not included in this schema are geometric items and representations, which are referenced from ISO 10303's generic resources, and the technology-specific definitions, which are defined in separate parts of ISO 14649.

This part of ISO 14649 cannot stand alone. An implementation needs in addition at least one technology-specific part (e.g. ISO 14649-11 for milling, ISO 14649-12 for turning).

Additionally, the schema uses machining features similar to ISO 10303-224 and ISO 10303-214. The description of process data is done using the EXPRESS language as defined in ISO 10303-11. The encoding of the data is done using ISO 10303-21.

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 286-1:1988, *ISO system of limits and fits — Part 1: Bases of tolerances, deviations, and fits*

ISO 286-2:1988, *ISO system of limits and fits — Part 2: Tables of standard tolerance grades and limit deviations for holes and shafts*

ISO 841, *Industrial automation systems and integration — Numerical control of machines — Coordinate system and motion nomenclature*

ISO 2806, *Industrial automation systems — Numerical control of machines — Vocabulary*

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

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

ISO 10303-21, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of exchange structure*

ISO 10303-41, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*

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

ISO 10303-43, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures*

ISO 10303-203, *Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol: Configuration controlled design*

ISO 10303-214:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes*

ISO 10303-224:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 224: Application protocol: Mechanical product definition for process planning using machining features*

ISO 10303-514:1999, *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation*

3 Terms and definitions

For the purposes of this document, the following terms, definitions and abbreviations apply.

2D:	Geometry in an xy-plane, where all the geometry's points have only x and y coordinates.
2½D machining:	Machining of a prismatic part. Typically, the workpiece is processed in several layers which are located perpendicular to the tool axis. In the EXPRESS listing of ISO 14649, the term “two5D” is used for entity and attribute names.
3D:	Geometry in three-dimensional space, where all points have x, y, and z coordinates.
freeform machining:	Machining of freeform surfaces. For this kind of machining, the tool has to move in at least three axes simultaneously while processing the workpiece. Sometimes five-axes milling machines are used to reach an optimised angle between the tool and the workpiece surface.
CAM	Computer Aided Manufacturing
CNC	Computer Numerical Control
EDM	Electrical Discharge Machining
EXPRESS:	The language described in ISO 10303-11.
EXPRESS-G:	The graphic representation of the EXPRESS language as described in ISO 10303-11.

Feature:	A geometric entity of a workpiece which has semantic significance. In the context of ISO 14649, manufacturing features are used.
SI	International system of units

4 General process data

4.1 Header and references

The following listing gives the header and the list of entities which are referenced within this schema.

```

SCHEMA machining_schema;

REFERENCE FROM approval_schema (* ISO 10303-41 *) (
    approval
);

REFERENCE FROM date_time_schema (* ISO 10303-41 *) (
    date_and_time
);

REFERENCE FROM measure_schema (* ISO 10303-41 *) (
    length_measure,
    parameter_value,
    plane_angle_measure,
    positive_length_measure,
    positive_ratio_measure,
    time_measure
);

REFERENCE FROM person_organization_schema (* ISO 10303-41 *) (
    person, address
);

REFERENCE FROM support_resource_schema (* ISO 10303-41 *) (
    identifier,
    label,
    text
);

REFERENCE FROM geometry_schema (* ISO 10303-42 *) (
    axis2_placement_3d,
    b_spline_curve,
    bounded_curve,
    bounded_pcurve,
    bounded_surface,
    cartesian_point,
    direction,
    elementary_surface,
    plane
);

REFERENCE FROM topology_schema (* ISO 10303-42 *) (
    face,
    open_shell
);

USE FROM geometric_model_schema (* ISO 10303-42 *) (
    block,

```

```

    right_circular_cylinder
);

REFERENCE FROM aic_topologically_bounded_surface (* ISO 10303-511 *) (
    advanced_face
);

REFERENCE FROM aic_advanced_brep (* ISO 10303-514 *) (
    advanced_brep_shape_representation
);

```

4.2 General types and definitions

4.2.1 Measure units

The types of units supported by ISO 14649 are SI units as well as derived or conversion based units as defined in ISO 10303-41. If no units are given then, the following units are assumed:

length_measure	millimetres [mm]
time_measure	seconds [s]
plane_angle_measure	degrees [°]
pressure_measure	Pascal [pa]
speed_measure	meters per second [m/s]
rot_speed_measure	revolutions per second [r/s]

4.2.1.1 Toleranced length measure

Length measure with tolerance.

```

ENTITY toleranced_length_measure;
    theoretical_size:      positive_length_measure;
    implicit_tolerance:    tolerance_select;
END_ENTITY;

```

theoretical_size: The theoretical length.

implicit_tolerance: The type of tolerance to apply to theoretical_size.

Note that all geometric properties of the workpiece are specified using toleranced_length_measure. If the NC controller has the ability to generate toolpaths or to make decisions about the tool used, it is the controller's responsibility to meet these tolerance requirements. On the other hand, data provided to the NC controller for explicit specification of movements will have no tolerances as the controller cannot do more than try to follow the given theoretical values to the best of its abilities. The same is true for offsets and other data referring to already toleranced dimensions.

4.2.1.1.1 Tolerance select

Select type offering different entities to describe tolerances for scalar values.

```

TYPE tolerance_select = SELECT(plus_minus_value, limits_and_fits);
END_TYPE;

```

4.2.1.1.2 Plus minus value

The plus_minus_value describes the upper and lower limits valid for a scalar dimension referencing this entity.

```

ENTITY plus_minus_value;
  upper_limit:      positive_length_measure;
  lower_limit:      positive_length_measure;
  significant_digits: INTEGER;
END_ENTITY;

```

upper_limit: The value of the tolerance that shall be added to the exact value to establish the maximum allowed value.

lower_limit: the value of the tolerance that shall be subtracted from the exact value to establish the minimum allowed value.

significant_digits: The number of decimal digits indicating the accuracy of the lower_bound and upper_bound values.

4.2.1.1.3 Limits and fits

A Limits_and_fits contains the necessary information to express a tolerance of the limits-and-fits system standardized by ISO 286.

```

ENTITY limits_and_fits;
  deviation:      length_measure;
  grade;;        length_measure
  its_fitting_type: OPTIONAL fitting_type;
END_ENTITY;

```

deviation: The difference between a measured actual size and the corresponding basic size as defined in ISO 286-1.

grade: The grade specifies the quality or the accuracy grade of a tolerance. The grade is based on the international standard tolerance grades IT01 to IT18 as defined in ISO 286-1.

fitting_type: Specification whether the tolerance declaration applies to a shaft or to a hole.

4.2.1.1.4 Fitting type

The enumeration used to specify the type of fitting.

```

TYPE fitting_type = ENUMERATION OF (shaft,hole);
END_TYPE;

```

4.2.1.2 Speed measure

A measure for a linear speed used for cutting speeds and feed rates.

```

TYPE speed_measure = REAL;
END_TYPE;

```

4.2.1.3 Rotational speed measure

A measure for a rotational speed. Positive values indicate rotation in the mathematical positive sense, i. e. counter-clockwise motion.

```

TYPE rot_speed_measure = REAL;
END_TYPE;

```

4.2.1.4 Pressure measure

A measure for pressure.

```

TYPE pressure_measure = REAL;
END_TYPE;

```

4.2.2 Other general types

4.2.2.1 Rotational direction

Enumeration used to identify the direction (sense) of the tool's rotation or of the direction of a circular movement. It should not be used in conjunction with `rot_speed_measure` which carries its own sense of rotation.

```
TYPE rot_direction = ENUMERATION OF (cw, ccw);
END_TYPE;
```

NOTE cw means clockwise, ccw means counter-clockwise.

4.2.2.2 Shape tolerance

Type for definition of shape tolerance.

```
TYPE shape_tolerance = length_measure;
END_TYPE;
```

4.3 Where to start: Project

Each part programme, i.e. data model, based on ISO 14649 must include exactly one top-level entity, called project. The project indicates the workplan to be executed upon interpretation of this model (as several workplans might be included), and it may also provide the workpiece(s) upon which actions are to be performed.

```
ENTITY project;
  its_id:          identifier;
  main_workplan:   workplan;
  its_workpieces:  SET [0:?] OF workpiece;
  its_owner:       OPTIONAL person_and_address;
  its_release:     OPTIONAL date_and_time;
  its_status:      OPTIONAL approval;
  (*
  Informal proposition:
  its_id shall be unique within the part programme.
  *)
END_ENTITY;
```

<code>its_id:</code>	The project's identifier. It shall be unique within the part programme.
<code>main_workplan:</code>	The top-level workplan in this model.
<code>its_workpieces:</code>	The workpieces upon which actions are to be performed.
<code>its_owner:</code>	Optional information on the owner of the project.
<code>its_release:</code>	Optional date and time reference of the project.
<code>its_status:</code>	Optional attribute to indicate the current status of the project.

4.3.1 Person and address

Entity includes data to name and reference a person, who for instance is responsible for creating a project.

```
ENTITY person_and_address;
  its_person:      person;
  its_address:     OPTIONAL address;
END_ENTITY;
```

4.4 What to machine: Workpiece and manufacturing feature

4.4.1 Workpiece

The workpiece entity contains the entire description of the workpiece, if available. This includes material, surface condition and geometric data. Each workpiece has only one surface condition and one material. Dependent on the conformance class the workpiece entity includes the raw part dimension only as an including box or cylinder, or, in a higher class, as a representation which can be a complete geometric model, e.g. the state after previous manufacturing operations.

```
ENTITY workpiece;
  its_id: identifier;
  its_material: OPTIONAL material;
  global_tolerance: OPTIONAL shape_tolerance;
  its_rawpiece: OPTIONAL workpiece;
  its_geometry: OPTIONAL advanced_brep_shape_representation;
  its_bounding_geometry: OPTIONAL bounding_geometry_select;
  clamping_positions: SET [0:?] OF cartesian_point;
END_ENTITY;
```

its_id:	The unique identification of a workpiece.
its_material:	The material attribute identifies the workpiece material. This data shall be used for determining the technological process parameters for the manufacturing process. It can be done by the machine operator or by an automatic feed rate/cutting condition selection (from a table or a data base on the CNC).
global_tolerance:	Tolerance for the workpiece, valid where no other tolerances are specified.
its_rawpiece:	The rawpiece geometry of the workpiece may be described here. A recursive description is used, i.e. the rawpiece is of type workpiece itself.
its_geometry:	An exact description of the final workpiece geometry according to ISO 10303-514.
its_bounding_geometry:	By this attribute the workpiece's bounding geometry might be defined as a box, a cylinder or a geometry according to the definition of the entity advanced_brep_shape_representation (ISO 10303-514).
clamping_positions:	Positions of the clamping device on the workpiece's surface.

All attribute's locations, directions and geometrical information are defined relatively to the workpiece's coordinate system.

4.4.1.1 Material

This entity is for identifying the workpiece material.

```
ENTITY material;
  standard_identifier: label;
  material_identifier: label;
  material_property: SET [0:?] OF property_parameter;
END_ENTITY;
```

standard_identifier:	The standard used for identifying the material. This can be a national standard or one used internally in the company.
material_identifier:	The name which identifies the material.
material_property:	The parameter which describes the properties of material. Since the demand for material properties varies, a generic type "property parameter" is used.

4.4.1.2 Property parameter

Generic property parameter which may be used to characterise any kind of property any kind of parameter might have. Subtypes are descriptive parameters (strings) and numeric parameters.

```
ENTITY property_parameter
  SUPERTYPE OF (ONEOF (descriptive_parameter, numeric_parameter));
  parameter_name: label;
END_ENTITY;
```

parameter_name: The name of the parameter to be described.

4.4.1.2.1 Descriptive parameter

A parameter description using a string to characterise the parameter.

```
ENTITY descriptive_parameter
  SUBTYPE OF (property_parameter);
  descriptive_string: text;
END_ENTITY;
```

descriptive_string: String value which describes the parameter.

4.4.1.2.2 Numeric parameter

A parameter description using a numeric value. Both the number and the unit shall be given.

```
ENTITY numeric_parameter
  SUBTYPE OF (Property_parameter);
  its_parameter_value: parameter_value;
  its_parameter_unit: label;
END_ENTITY;
```

its_parameter_value: The value which describes the parameter.

its_parameter_unit: The units associated with the value.

4.4.1.3 Bounding geometry select

This type offers three different entities to describe the bounding geometry of a workpiece. Depending on the conformance class the bounding geometry is described as a block or a cylinder using the geometric items of ISO 10303-42 or as a complex shape conformable to ISO 10303-514. All coordinates and directions given in bounding_geometry are defined in the given workpiece's coordinate system.

```
TYPE bounding_geometry_select = SELECT (
  block, right_circular_cylinder, (* ISO 10303-42 *)
  advanced_brep_shape_representation (* ISO 10303-514*)
);
END_TYPE;
```

4.4.2 Manufacturing feature

This entity is the supertype of all manufacturing features. When considering 2½D-manufacturing, features are of type two5D_manufacturing_feature and may be holes, pockets, etc. When considering freeform manufacturing, regions are used in the same sense.

The manufacturing feature describes the feature as such, e.g. by its geometric properties. It does not give any instruction on how the workpiece is manufactured. Such process related information is contained only in the operations and depends on the manufacturing method. The methods are described in the technology specific parts of ISO 14649. For example, plunging into a pocket, roughing the pocket and finishing the pocket bottom

may all be individual operations, as well as the finishing of a portion of a freeform surface. For the concept of operations, please refer to section 4.7.1.

```
ENTITY manufacturing_feature
  ABSTRACT SUPERTYPE OF (ONEOF(region, two5D_manufacturing_feature,
    transition_feature));
  its_id: identifier;
  its_workpiece: workpiece;
  its_operations: SET [0:?] OF machining_operation;
END_ENTITY;
```

its_id: Each feature has an unique identifier.

its_workpiece: The workpiece which the feature is part of.

its_operations: A set of all operations associated with the feature required for the manufacturing of the feature. Note that the operations are not necessarily executed immediately after each other. Only the workplan determines the final order of operations of all operations included in a given programme, and manufacturing will typically not occur feature by feature but rather according to technological criteria like minimised tool change. However, it should be guaranteed by the CAM system (or by the controller, if the order of execution is determined by an intelligent CNC) that the order given in this attribute is never violated.

4.5 Catalogue of manufacturing features

4.5.1 Region

The region is the equivalent of a feature in freeform machining. It describes a bounded area of the final workpiece surface to which the associated operations will be applied. Note that for freeform surfaces the regions for different operations (e.g. roughing, finishing) may have different shapes based upon technological decisions.

```
ENTITY region
  ABSTRACT SUPERTYPE OF (ONEOF (region_surface_list, region_projection,
    topological_region))
  SUBTYPE OF (manufacturing_feature);
  feature_placement: OPTIONAL axis2_placement_3d;
END_ENTITY;
```

feature_placement: The placement of the feature relative to the workpiece co-ordinate system. The placement is a translation and/or a rotation which transforms the origin of the workpiece co-ordinate system origin into the origin of the feature's local co-ordinate system. If no feature_placement is given, the region will use workpiece co-ordinates. Regarding coordinate systems see also Section 4.6.4.1.2.

4.5.1.1 Region projection

A type of bounded region generated by projecting a closed curve on a surface. Thus, a region can easily be defined on an existing surface description. The region projection applies to any workpiece surface it hits when moving in space along proj_dir. This is a very simplistic region definition for situations where the use of topology is not desired.

```
ENTITY region_projection
  SUBTYPE OF (region);
  proj_curve : bounded_curve;
  proj_dir : direction;
  depth: toleranced_length_measure;

END_ENTITY;
```

proj_curve:	A curve in space, used for specifying the boundary for a surface through its projection on the surface. It must be a closed curve and will usually be in one plane.
proj_dir:	Direction used for projecting the above curve on the surface.
depth:	The depth is a positive scalar value. It describes the distance of material removal into negative z-direction of the region's local coordinate system. The depth is measured from the raw piece's surface, onto which the proj_curve has been projected, to the bottom to be machined. The depth is equal for the whole region. In case the bottom's shape shall not be equal to the raw piece's surface, the entity pocket and its subtypes have to be used.

4.5.1.2 Region surface list

A type of region specified by a list of surfaces. This allows for the most general description of regions. The region is bounded by the borders of the surfaces it encloses.

```
ENTITY region_surface_list
  SUBTYPE OF (region);
  surface_list :      LIST [1:?] OF bounded_surface;
END_ENTITY;
```

surface_list: List of general surfaces, allowing for various descriptions.

4.5.1.3 Topological region

A type of region specified by a topological representation. Only faces of type advanced_face are allowed as they are certain to carry their own geometric representation.

```
ENTITY topological_region
  SUBTYPE OF (region, open_shell);
WHERE
  WR1: SIZEOF(QUERY(it <* SELF.cfs_faces |
    NOT('MACHINING_SCHEMA.ADVANCED_FACE' IN TYPEOF(it)))) = 0;
END_ENTITY;
```

4.5.2 Two5D manufacturing feature

The entity two5D_manufacturing_feature is the abstract supertype of all 2½D features. This structure is defined in close resemblance to ISO 10303-224.

```
ENTITY two5D_manufacturing_feature
  ABSTRACT SUPERTYPE OF (ONEOF(machining_feature, replicate_feature,
    compound_feature))
  SUBTYPE OF (manufacturing_feature);
  feature_placement:      axis2_placement_3d;
END_ENTITY;
```

feature_placement: The placement of the feature relative to the workpiece co-ordinate system. The placement is a translation and/or a rotation which transforms the origin of the workpiece co-ordinate system origin into the origin of the feature's local co-ordinate system. For information on coordinate systems refer to Section 4.6.4.1.2. If the manufacturing feature is part of a compound_feature, then its placement is defined relative to the compound_feature's origin.

4.5.3 Machining feature

The entity machining_feature is the abstract supertype of all features used for feature based 2½D machining. In this part the following features are foreseen as they are used in different technology specific parts of ISO 14649: planar face, pocket, slot, step, hole, generic feature, and compound feature. The features are defined in close resemblance to ISO 10303-224.

2½D machining is characterised by the fact that most tool movements occur only in the xy plane while the z-axis is preset to a certain depth in order to remove a layer of material. For this reason, all machining features have a depth.

All feature geometry, e.g. the contour describing the outline of a pocket, is described in a local xyz co-ordinate system. The definition of the coordinate system's orientation is given in the machining method specific parts. The surrounding surface of the workpiece and the definition of the planar contour of the feature are assumed to lie in the xy plane ($z = 0$). The material is assumed to be in negative z direction of the xy-plane. In other words, a positive depth within a machining operation requires the tool to advance in negative z direction into the material's direction.

As ISO 14649 allows for a 3D description of the overall workpiece geometry, the inherited attribute `feature_placement` specifies the actual location of the feature within the workpiece co-ordinate system.

```
ENTITY machining_feature
ABSTRACT SUPERTYPE OF (ONEOF(planar_face, pocket, slot, step, round_hole,
    toolpath_feature, profile_feature, boss, spherical_cap, rounded_end,
    thread))
SUBTYPE OF (two5D_manufacturing_feature);
depth: elementary_surface;
END_ENTITY;
```

depth: The depth of the feature is described by a plane which includes the lowest points of the feature ($z < 0$) measured in the feature's local cartesian-coordinate system. If the depth is not an orthogonal plane to the z-axis this implies, that the bottom of the feature is inclined. Depending on the explicit features there can be described more complex bottom shapes.

4.5.4 Planar face

The planar face is used to describe machining of the outer face of a workpiece. The geometry of the planar face is given through the boundary and the depth (defined by the supertype). The depth denotes the the bottom of the material that needs to be taken away from the workpiece to reach the final shape of the feature. The elementary surface describing the depth can be inclined or orthogonal to the feature's local z-axis. The planar face may also have one or more bosses where no cutting will be done.

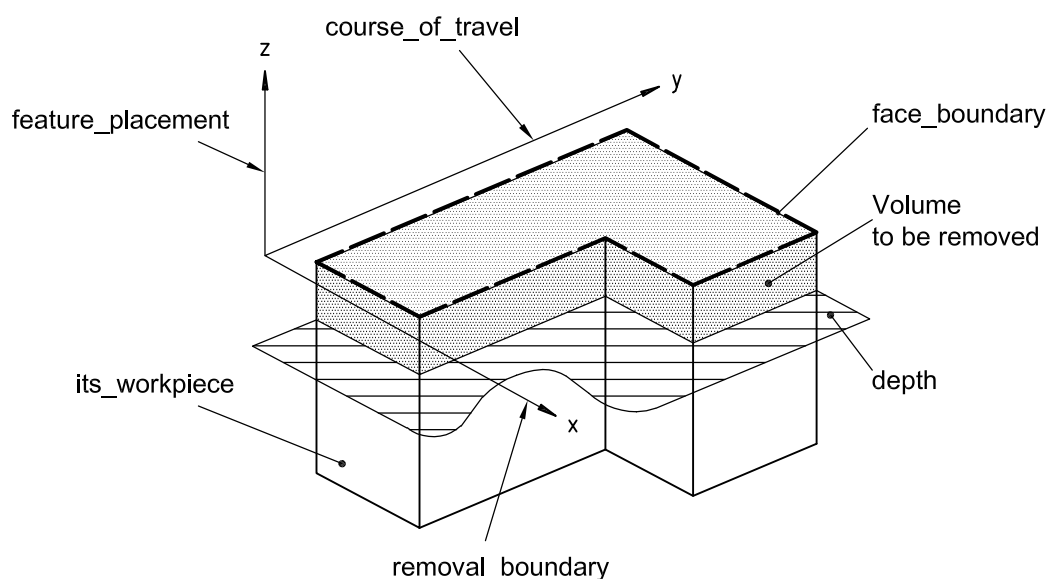


Figure 1 — Planar face

```

ENTITY planar_face
  SUBTYPE OF (machining_feature);
  course_of_travel:      linear_path;
  removal_boundary:      linear_profile;
  face_boundary:         OPTIONAL closed_profile;
  its_boss:              SET [0:?] OF boss;
(*
Informal propositions:
- The entire profile lies in the local xy plane.
- The profile is not self-intersecting.
*)
END_ENTITY;

```

removal_direction:	The direction of material removal.
course_of_travel:	A straight line with magnitude and direction.
removal_boundary:	A line with direction and magnitude that when swept along a path defines the area on a workpiece for volume removal.
face_boundary:	The complete or partial outside final shape of the workpiece after the planar cut has been applied.
its_boss:	An optional list of entities which describe one or more bosses. A boss of a planar_face defines a part of the planar_face which is not cut during manufacturing of the planar_face.

4.5.5 Pocket

This is the abstract supertype for different implicit and explicit pockets. Open and closed pockets are derived from this supertype. The geometry of the pocket is defined by its contour on the outer face of the workpiece and its depth. A pocket may possess one or more bosses.

```

ENTITY pocket
  ABSTRACT SUPERTYPE OF (ONEOF(closed_pocket, open_pocket))
  SUBTYPE OF (machining_feature);
  its_boss:      SET [0:?] OF boss;
  slope:         OPTIONAL plane_angle_measure;
  bottom_condition: pocket_bottom_condition;
  planar_radius:  OPTIONAL toleranced_length_measure;
  orthogonal_radius:  OPTIONAL toleranced_length_measure;
END_ENTITY;

```

its_boss:	Optional list of bosses. This defines one or more parts of the pocket which are not cut during manufacturing of the pocket. With the cutting of the pocket, the boss is cut simultaneously.
slope:	Optional angle of the border of the pocket measured against the local z axis. Default is 0 degrees.
bottom_condition:	Possible bottom conditions of the pocket.
planar_radius:	The planar radius of a fillet, if existing within a pocket. This may be used to check against the radius of the tool tip.
orthogonal_radius:	The orthogonal radius of a fillet, if existing within a pocket. This may be used to check against the radius of the tool.

4.5.5.1 Closed pocket

Derived from the class pocket. A closed pocket is a pocket which is surrounded by material everywhere along its circumference. Its feature boundary is given by a closed profile.

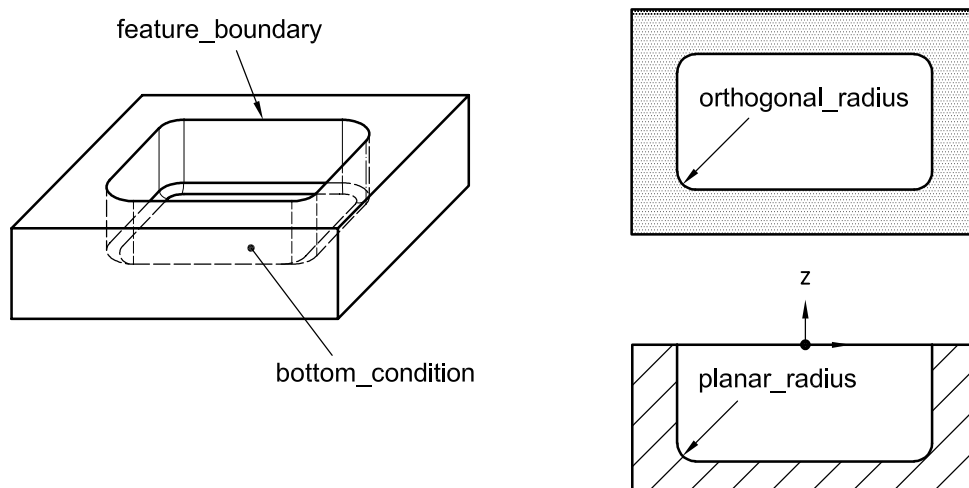


Figure 2 — Closed pocket

```

ENTITY closed_pocket
  SUBTYPE OF (pocket);
  feature_boundary:    closed_profile;
  (*
  Informal propositions:
  - The entire profile lies in the feature's local xy plane.
  - The profile is closed and not self-intersecting.
  *)
END_ENTITY;

```

feature_boundary: The shape or outline that describes the upper edge of the pocket. It is an enclosed area that has a completely closed profile. The profile specifies the area required by a closed pocket.

4.5.5.2 Open pocket

Derived from the class pocket. An open pocket is a pocket which is not a closed pocket. Its feature boundary is given by a wall contour.

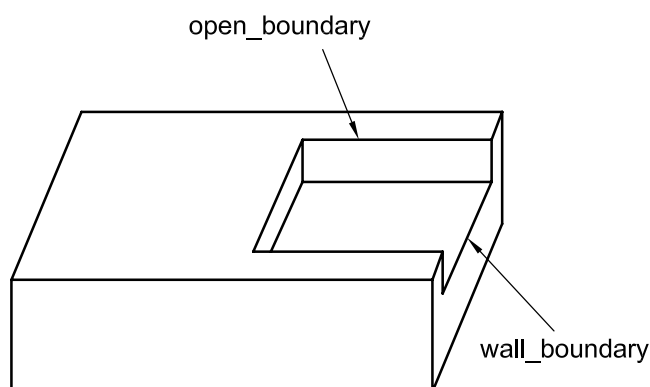


Figure 3 — Open pocket

```

ENTITY open_pocket
  SUBTYPE OF (pocket);
  open_boundary:      open_profile;
  wall_boundary:      OPTIONAL open_profile;
  (*)
  Informal propositions:
  - The entire open_boundary profile lies in the local xy plane.
  - The open_profiles are not self-intersecting.
  - Together the two open_profiles form a closed profile.
  - wall_boundary is for information only.
  *)
END_ENTITY;

```

open_boundary: The outline or shape that forms the upper edge of the open_pocket. The profile specifies the area required by open pocket. When travelling along the profile based as defined by its sense, the material lies on the left side of the profile. The open_boundary lies within the features xy-plane ($z=0$).

wall_boundary: The outline or shape that forms the side edge of the open_pocket. It forms a closed profile together with open_boundary. Note that this contour will be most likely defined implicitly by the selected tool and the fillet options inherited from machining_feature. If given, it will be informative only.

4.5.5.3 Pocket bottom condition

This entity describes the shape of the bottom of a pocket.

```

ENTITY pocket_bottom_condition
  ABSTRACT SUPERTYPE OF
    (ONEOF (through_pocket_bottom_condition, planar_pocket_bottom_condition,
      radiused_pocket_bottom_condition, general_pocket_bottom_condition));
END_ENTITY;

```

4.5.5.3.1 Through pocket bottom condition

The pocket extends fully through the workpiece.

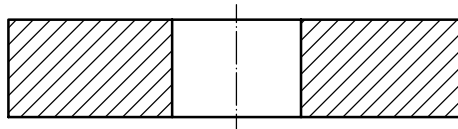


Figure 4 — Through pocket bottom condition

```

ENTITY through_pocket_bottom_condition
  SUBTYPE OF (pocket_bottom_condition);
END_ENTITY;

```

4.5.5.3.2 Planar pocket bottom condition

The pocket has a plane bottom.

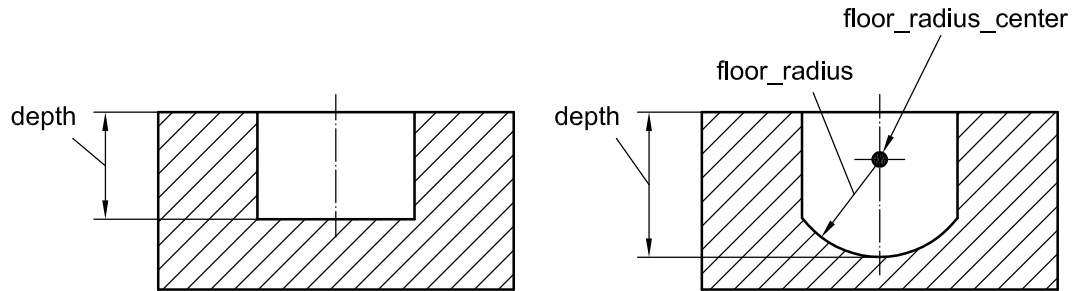


Figure 5 — Planar (left) and radiused (right) pocket bottom condition

```
ENTITY planar_pocket_bottom_condition
  SUBTYPE OF (pocket_bottom_condition);
END_ENTITY;
```

4.5.5.3.3 Radiused pocket bottom condition

The pocket has a radiused bottom. As opposed to the use of the attribute planar_radius of the machining feature, which may be used to specify a fillet, the entire bottom surface will be continuously curved.

```
ENTITY radiused_pocket_bottom_condition
  SUBTYPE OF (pocket_bottom_condition);
  floor_radius_center: cartesian_point;
  floor_radius: toleranced_length_measure;
END_ENTITY;
```

floor_radius_center: Center of the radius of the pocket floor. It is defined within the features coordinate system.

floor_radius: Radius of the pocket floor. The radius creates a portion of a sphere on the pocket floor. Concave or convex behaviour depends upon the position of floor_radius_center. Note that the diameter of the sphere must span at least the perimeter of the pocket floor, or the result will be undefined.

Note that the depth of the pocket is determined by the combination of floor_radius_center and floor_radius

4.5.5.3.4 General pocket bottom condition

Any bottom condition not covered by planar_bottom_condition or through_bottom_condition. General pocket bottom condition defines a free-form surface at the bottom of a pocket.

Similar to the compound feature, the region referenced by the attribute shape is not allowed to be associated with machining operations of its own. All machining operations to manufacture the pocket must be defined by the attribute its_operations of the pocket itself.

```
ENTITY general_pocket_bottom_condition
  SUBTYPE OF (pocket_bottom_condition);
  shape: region;
WHERE
  WR1: SIZEOF(shape\manufacturing_feature.its_operations) = 0;
END_ENTITY;
```

shape: Description of the pocket floor as 3D surface in local xyz co-ordinates. Note that the feature_placement attribute of shape will specify its position relative to the local co-ordinate system of the pocket, not relative to the workpiece co-ordinates.

The pocket's attribute depth should specify the maximum depth of the pocket. Note that the actual depth of the pocket will be determined by the shape of the surface which is defined in 3D space. It will override any conflicting data given in the attribute depth.

4.5.6 Slot

The entity slot is closely related to the entity pocket. Generally speaking, a slot is a special kind of pocket. ISO 10303-224 lists slot as a separate entity. To be compatible, the entity slot is also included in this schema. The attribute `course_of_travel` describes the location and extension of the slot. Typically, the slot will be manufactured by a single sweep of a tool along the course of travel. In this case, the width of the slot equals the shape of the tool. In case of a milling operation the shape is given by the tool's diameter. If a workingstep which manufactures this slot calls for a smaller tool, more than one cut will have to be made.

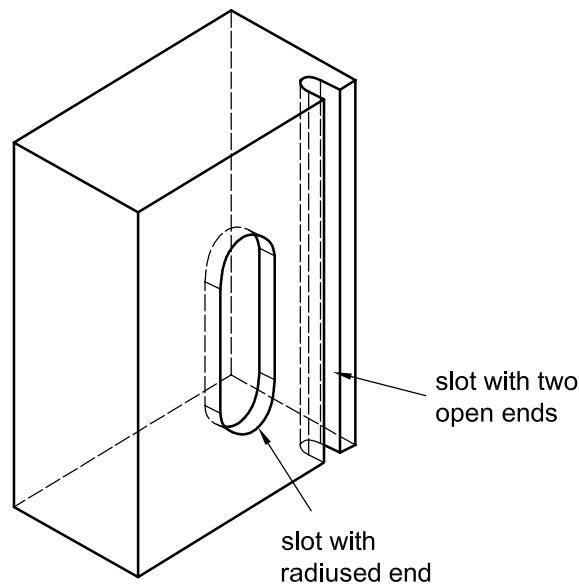


Figure 6 — Different examples of slots

```
ENTITY slot
  SUBTYPE OF (machining_feature);
  course_of_travel:   travel_path;
  swept_shape:       open_profile;
  end_conditions:     LIST[0:2] OF slot_end_type;
WHERE
  WR1: (
    ( SIZEOF(QUERY (it <* SELF.end_conditions |
      ('MACHINING_SCHEMA.LOOP_SLOT_END_TYPE' IN TYPEOF(it)) )) = 1)
    AND
    (SIZEOF(end_conditions) = 1)
  )
  OR
  (SIZEOF(end_conditions) <> 1);

(*
Informal propositions:
- The entire travel_path lies in the local xy plane.
- The travel_path is not self-intersecting.
*)
END_ENTITY;
```

course_of_travel: Center line of the slot. The tool is moved along this path to achieve the manufacturing of the slot. As with the entity pocket, the upper edge is given.

swept_shape: The (contoured) cross-section generated by a the tool, required for the selection of the proper tool For simple rectangular slot profiles, a square_u_profile should be specified giving only the width of the slot.

end_conditions: For a slot closed on one side, end conditions may be given here. The sequence is oriented conformable to the direction of the course_of_travel.

4.5.6.1 Slot end type

Supertype of slot end types.

```
ENTITY slot_end_type
  ABSTRACT SUPERTYPE OF (ONEOF (woodruff_slot_end_type, radiused_slot_end_type,
    flat_slot_end_type, loop_slot_end_type, open_slot_end_type));
END_ENTITY;
```

4.5.6.1.1 Woodruff slot end type

The end of slot shall be a radius tangent to the slot bottom and curved upward about an axis.

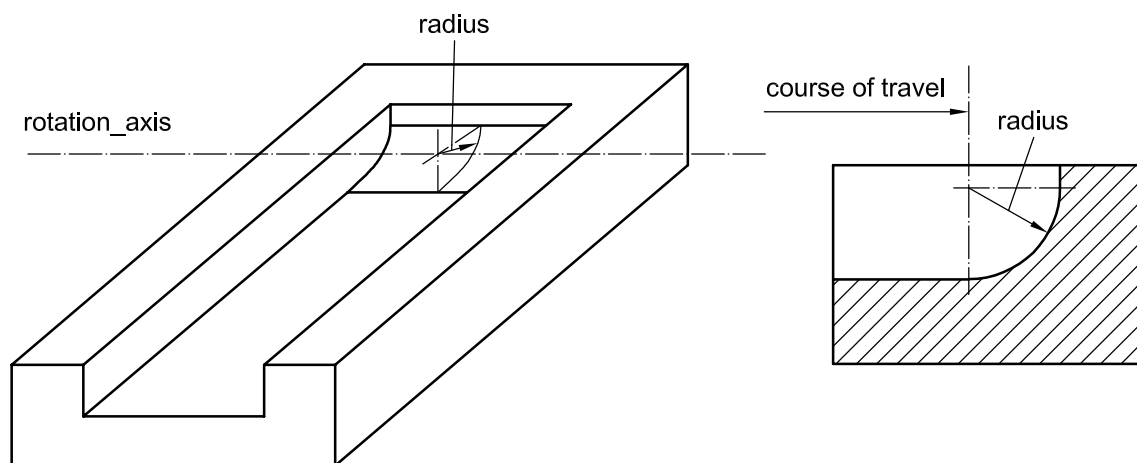


Figure 7 — Woodruff type slot end type

```
ENTITY woodruff_slot_end_type
  SUBTYPE OF (slot_end_type);
  radius:          toleranced_length_measure;
END_ENTITY;
```

radius: Radius of the slot end type. The radius continues tangential from the bottom of the slot. It starts at the bottom of the slot where the course_of_travel ends. The radius does not necessarily equal to the depth of the slot.

4.5.6.1.2 Radiused slot end type

The end of the slot consists of an arc. The diameter equals the width of the slot. The center of the arc is identical with the end point of course_of_travel.

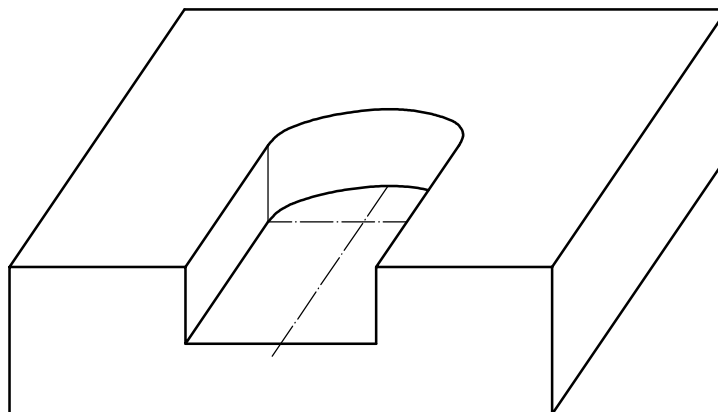


Figure 8 — Radiused slot end type

```
ENTITY radiused_slot_end_type
  SUBTYPE OF (slot_end_type);
END_ENTITY;
```

4.5.6.1.3 Flat slot end type

The end of the slot consists of a flat line with two arcs connecting the end to the sides of the slot. The radii of the two arcs are given. When traversing the slot along its center line from start to finish, the radius on the left side of the center line is *corner_radius1*, the radius on the right side of the center line is *corner_radius2*. The end point of *course_of_travel* is in the flat end of the slot.

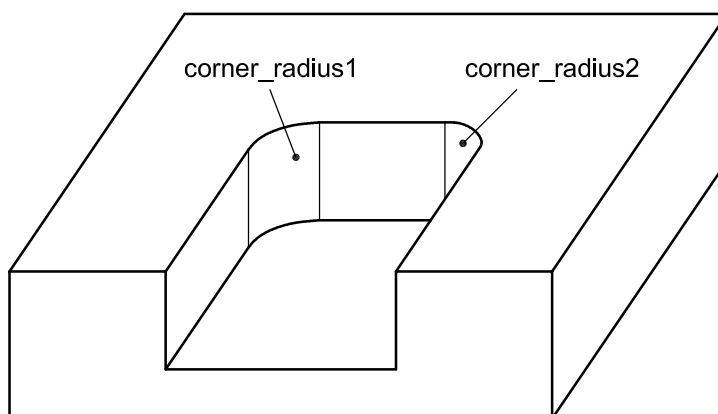


Figure 9 — Flat slot end type

```
ENTITY flat_slot_end_type
  SUBTYPE OF (slot_end_type);
  corner_radius1 :      toleranced_length_measure;
  corner_radius2 :      toleranced_length_measure;
END_ENTITY;
```

corner_radius1: Radius of the first arc.
corner_radius2: Radius of the second arc.

4.5.6.1.4 Loop slot end type

The slot forms a loop. The attribute depth is used to determine the scalar profoundness of the slot. The profoundness is equal to the smallest distance between the *course_of_travel* and the planar face described by the feature's attribute "depth".

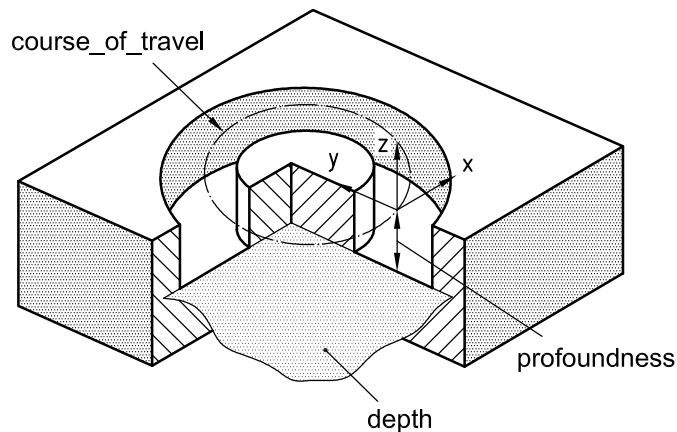


Figure 10 — Loop slot end type

```
ENTITY loop_slot_end_type
  SUBTYPE OF (slot_end_type);
END_ENTITY;
```

4.5.6.1.5 Open slot end type

The end of the slot is open.

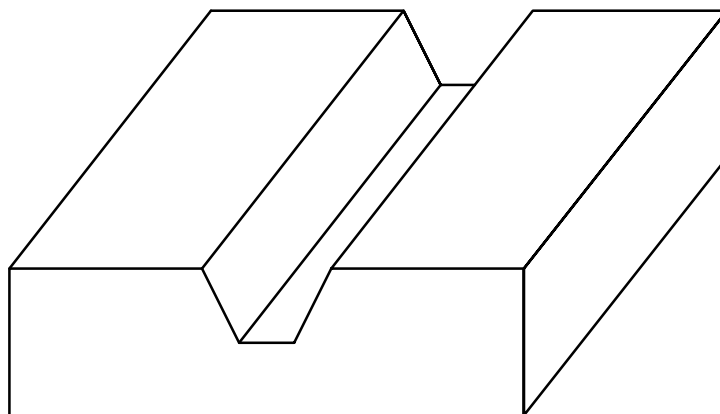


Figure 11 — Open slot end condition

```
ENTITY open_slot_end_type
  SUBTYPE OF (slot_end_type);
END_ENTITY;
```

4.5.7 Step

A step (or shoulder) is a volume of material removed from the top and the sides of the workpiece. Like an open pocket, its contour is open to its sides. The part of the V-profile describing the step's bottom lies in the elementary surface defined by the feature's attribute depth. The step may have one or more bosses as given by the optional attribute.

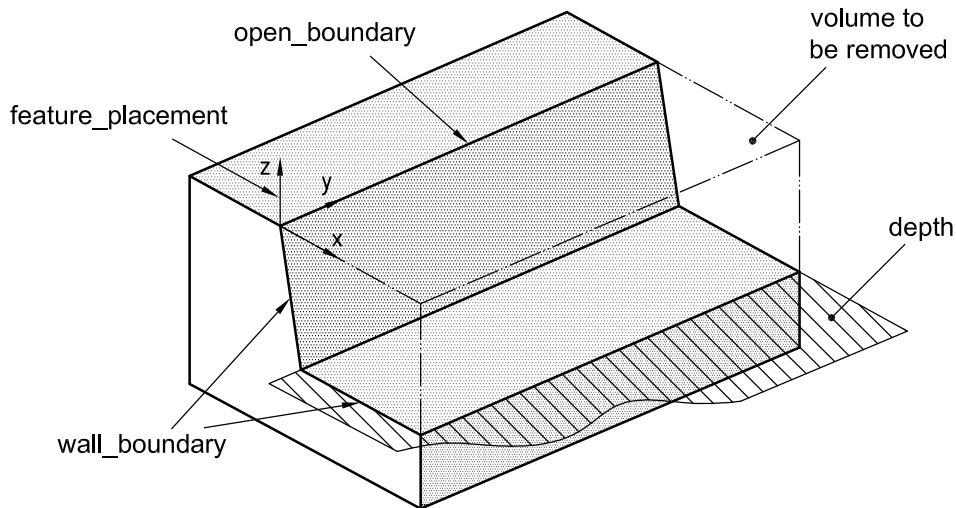


Figure 12 — Step

```
ENTITY step
  SUBTYPE OF (machining_feature);
  open_boundary:      linear_path;
  wall_boundary:      OPTIONAL vee_profile;
  its_boss:           SET[0:?] OF boss;
  (*
  Informal propositions:
  - The entire linear_path lies in the same plane.
  *)
END_ENTITY;
```

open_boundary: The outline or shape that forms the upper edge of the step. When travelling along the curve based as defined by its sense, the material is to the left of the curve.

wall_boundary: The outline or shape that forms the side edge of the step, i. e. the “profile” of the cut.

its_boss: A step may have one or more bosses.

4.5.8 Profile feature

A profile feature is a volume of material removed from the boundary shape of a workpiece. This is an abstract supertype of general outside profile and shape profile.

```
ENTITY profile_feature
  ABSTRACT SUPERTYPE OF (ONEOF (general_outside_profile, shape_profile))
  SUBTYPE OF (machining_feature);
  profile_swept_shape : linear_path;
END_ENTITY;
```

profile_swept_shape: An 2D line, when combined with a profile, which creates the shape of the profile feature. The placement of the linear path shall be the same as the profile feature. The direction of linear shall be with the Z-axis toward the direction of travel of the profile boundary.

4.5.8.1 General outside profile

A general outside profile is the removal volume of arbitrary shape from the outside shape of a workpiece. It may remove material from the entire outside shape or some portion of the shape. The contour of the shape is given by the attribute `feature_boundary`.

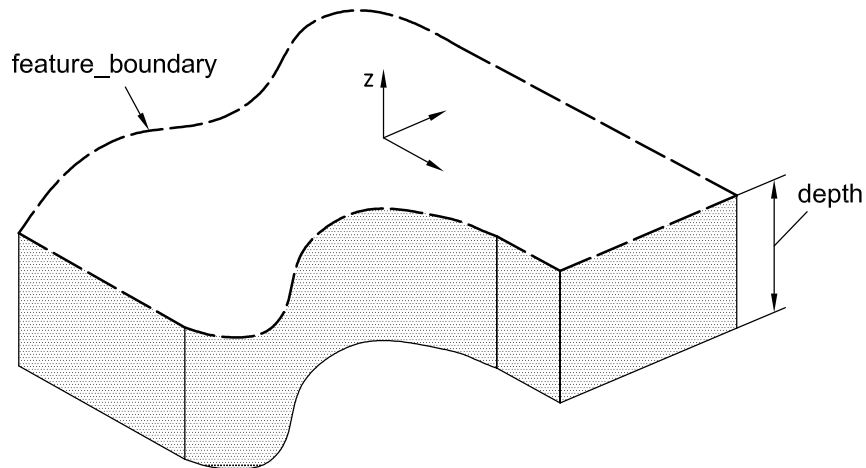


Figure 13 — General outside profile

```
ENTITY general_outside_profile
  SUBTYPE OF (profile_feature);
  feature_boundary: profile;
  (*
  Informal propositions:
  - The entire profile lies in the local xy plane.
  - The profile is not self-intersecting.
  *)
END_ENTITY;
```

feature_boundary: The contour of the profile to be followed by the tool. Note that this attribute describes the profile itself, not the tool path. When travelling along the profile based as defined by its sense, the material lies to the left of the profile.

4.5.8.2 Shape profile

A shape profile is the removal volume of shaped profile from the boundary shape of a workpiece. The bottom of the boundary shape is limited by a floor condition. This is an abstract supertype of general shape profile, partial circular shape profile, circular closed shape profile, rectangular open shape profile, and rectangular closed shape profile.

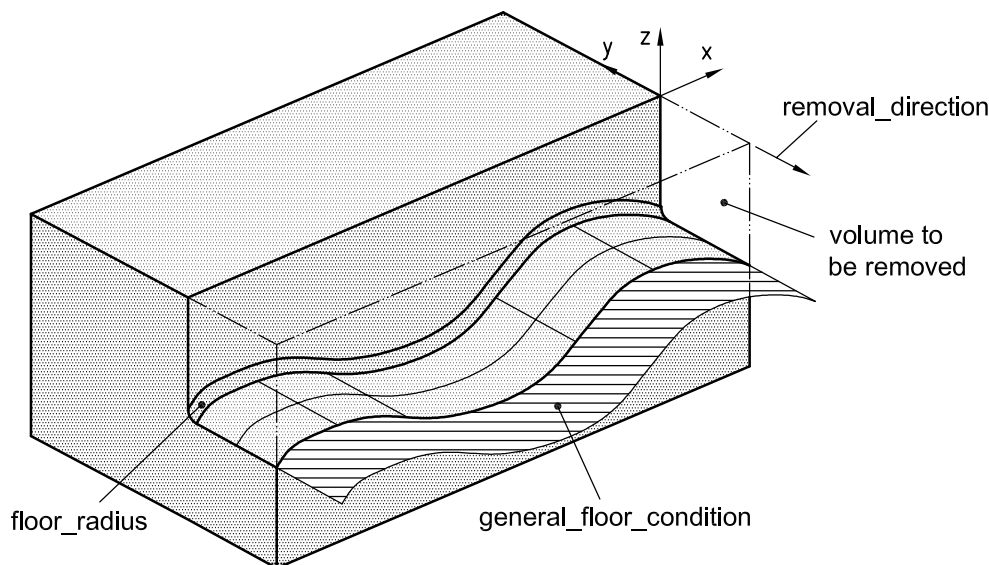


Figure 14 — Shape Profile

```

ENTITY shape_profile
  ABSTRACT SUPERTYPE
    OF(ONEOF(general_shape_profile,partial_circular_shape_profile,
      circular_closed_shape_profile,rectangular_open_shape_profile,
      rectangular_closed_shape_profile))
  SUBTYPE OF (profile_feature);
    floor_condition:      profile_select;
    removal_direction:    direction;
END_ENTITY;

```

floor_condition: Specification of the shape of the bottom.

removal_direction: A vector that points in the general direction away from the material.

4.5.8.2.1 Profile select

The floor condition of shape profile is either a through profile floor or a through profile floor.

```

TYPE profile_select =
  SELECT (through_profile_floor, profile_floor);
END_TYPE;

```

4.5.8.2.2 Through profile floor

A through profile floor describes the bottom condition of a shape profile which is open.

```

ENTITY through_profile_floor;
END_ENTITY;

```

4.5.8.2.3 Profile floor

A profile floor describes the bottom condition of a shape profile which may be flat or any arbitrary shape.

```

ENTITY profile_floor
  ABSTRACT SUPERTYPE OF(ONEOF(general_profile_floor, planar_profile_floor));
    floor_radius :      OPTIONAL numeric_parameter;
    start_or_end :      BOOLEAN;
END_ENTITY;

```

floor_radius: The radius of curvature for an arc between the bottom and the sides.

start_or_end: If true, the `profile_floor` is positioned at the end of a `shape_profile`. If false, its is at the start of the `shape_profile`.

4.5.8.2.4 General profile floor

A `general_profile_floor` is a type of `profile_floor` that specifies an enclosed area bounded by an arbitrary shape.

```
ENTITY general_profile_floor
  SUBTYPE OF (profile_floor);
  floor : face;
END_ENTITY;
```

floor: Specification of the face at the bottom of a `shape_profile`, adjacent to all the `shape_profile` walls.

4.5.8.2.5 Planar profile floor

A `planar_profile_floor` is a type of a `profile_floor` that characterises the bottom of a `shape_profile` which is flat.

```
ENTITY planar_profile_floor
  SUBTYPE OF (profile_floor);
  floor : plane;
END_ENTITY;
```

floor: Specification of a planar face at the bottom of a `shape_profile`.

4.5.8.2.6 General shape profile

A `general_shape_profile` is a type of `shape_profile` that is a volume of arbitrary shape which defines a portion of the workpiece.

```
ENTITY general_shape_profile
  SUBTYPE OF (shape_profile);
  profile_boundary : profile;
END_ENTITY;
```

profile_boundary: Specification of the outline of the `shape_profile` feature. It defines an area that may or may not be entirely enclosed.

4.5.8.2.7 Partial circular shape profile

A `partial_circular_shape_profile` is a type of `shape_profile` that defines a volume that is not enclosed on all sides.

```
ENTITY partial_circular_shape_profile
  SUBTYPE OF (shape_profile);
  open_boundary : partial_circular_profile;
END_ENTITY;
```

open_boundary: Specification of the outline of the `shape_profile` feature. It defines an area that shall be circular and shall not be enclosed.

4.5.8.2.8 Circular closed shape profile

A `circular_closed_shape_profile` is a type of `shape_profile` that defines a completely enclosed volume. This may have a thread.

```
ENTITY circular_closed_shape_profile
SUBTYPE OF (shape_profile);
  closed_boundary :      circular_closed_profile;
END_ENTITY;
```

`closed_boundary`: Specification of the outline of the `shape_profile` feature. It defines an area that shall be enclosed and circular.

4.5.8.2.9 Rectangular open shape profile

A `rectangular_open_shape_profile` is a type of `shape_profile` that is an open profile with opposite sides that are of equal length and with one side that does not make contact with the workpiece.

```
ENTITY rectangular_open_shape_profile
SUBTYPE OF (shape_profile);
  open_boundary :      square_u_profile;
END_ENTITY;
```

`open_boundary`: Specification of the outline or shape that is an enclosed area that is open on one side.

4.5.8.2.10 Rectangular closed shape profile

A `rectangular_closed_shape_profile` is a type of `shape_profile` that is an enclosed volume with opposite sides that are equal in length.

```
ENTITY rectangular_closed_shape_profile
SUBTYPE OF (shape_profile);
  closed_boundary :      rectangular_closed_profile;
END_ENTITY;
```

`closed_boundary`: Specification of the outline or shape that is an enclosed area that has a completely closed profile.

4.5.9 Round hole

This entity defines both holes and threaded holes. The `feature_placement` of a hole is the center point at the surface, i. e. the hole is located at $x = y = 0$ in the local co-ordinate system. The specified (positive) depth causes the tool to advance into the hole in negative z direction.

Note that the bottom of the hole is not considered in the hole's depth.

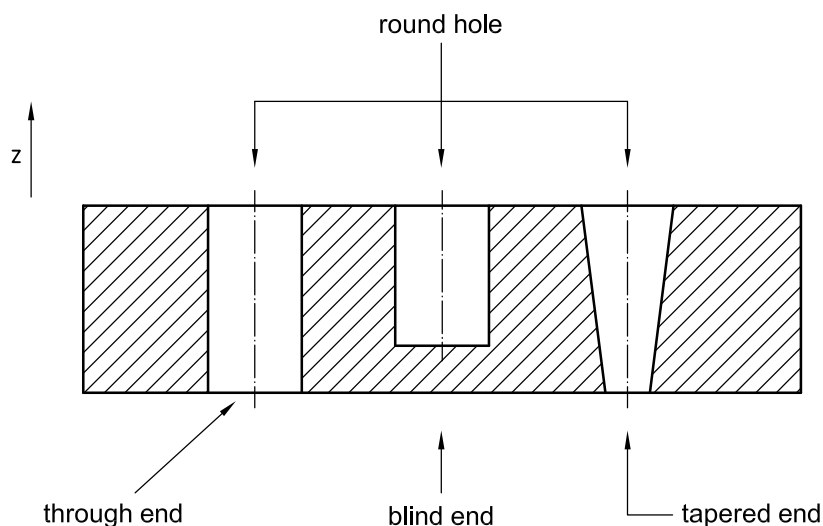


Figure 15 — Hole types

```

ENTITY round_hole
  SUBTYPE OF (machining_feature);
  diameter:          toleranced_length_measure;
  change_in_diameter:  OPTIONAL taper_select;
  bottom_condition:   hole_bottom_condition;
END_ENTITY;

```

diameter: The diameter of the hole. The diameter is measured in the xy-plane where $z = 0$.

change_in_diameter: An optional parameter used to specify holes with a taper.

bottom_condition: Specification of the bottom of a hole.

4.5.9.1 Taper select

The taper select indicates the manner by which a taper is described. A taper may be described either by diameter or by angle.

```

TYPE taper_select = SELECT (diameter_taper, angle_taper);
END_TYPE;

```

4.5.9.1.1 Diameter taper

Entity to describe a taper by its final diameter.

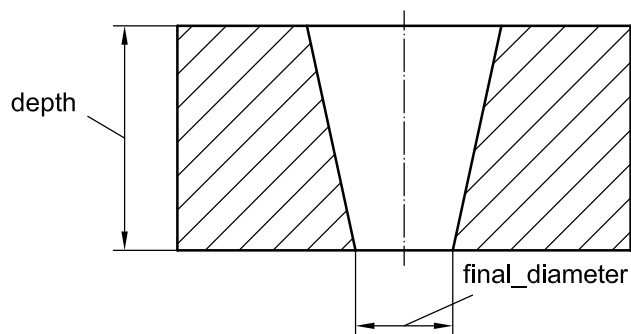


Figure 16 — Diameter taper

```
ENTITY diameter_taper;
  final_diameter:      toleranced_length_measure;
END_ENTITY;
```

final_diameter: The final diameter of the tapered hole at the indicated depth.

4.5.9.1.2 Angle taper

Entity to describe a taper by its half angle.

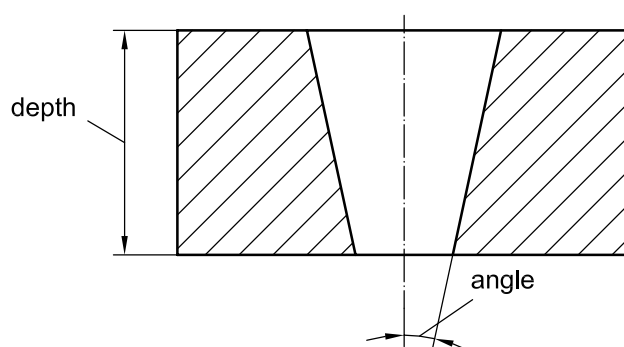


Figure 17 — Angle taper

```
ENTITY angle_taper;
  angle:      plane_angle_measure;
END_ENTITY;
```

angle: The angle of the tapered hole.

4.5.9.2 Hole bottom condition

Abstract supertype for the description of a bottom of a hole.

```
ENTITY hole_bottom_condition
  ABSTRACT SUPERTYPE OF (ONEOF (blind_bottom_condition,
    through_bottom_condition));
END_ENTITY;
```

4.5.9.2.1 Through bottom condition

Entity which describes a hole bottom which is open.

```
ENTITY through_bottom_condition
  SUBTYPE OF (hole_bottom_condition);
END_ENTITY;
```

4.5.9.2.2 Blind bottom condition

Supertype to describe different types of blind bottom conditions. The bottom may break through the bottom of the workpiece, but is not entirely open. The depth of the hole is the length of the cylindrical section of the hole.

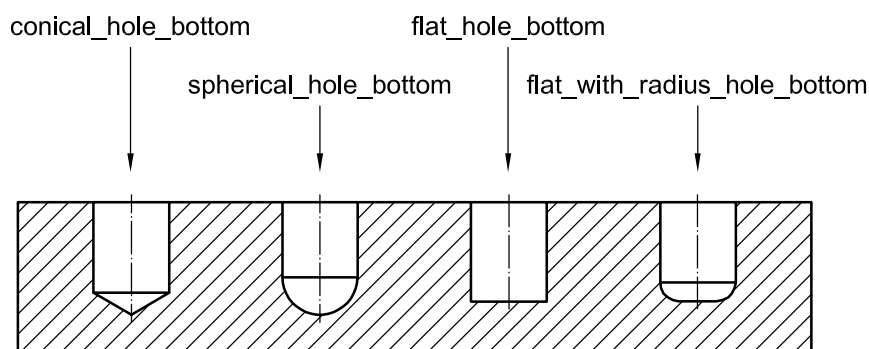


Figure 18 — Bottom conditions of holes

```
ENTITY blind_bottom_condition
  ABSTRACT SUPERTYPE OF (ONEOF(flat_hole_bottom, flat_with_radius_hole_bottom,
    spherical_hole_bottom, conical_hole_bottom))
  SUBTYPE OF (hole_bottom_condition);
END_ENTITY;
```

4.5.9.2.2.1 Flat hole bottom

A hole with a flat bottom.

```
ENTITY flat_hole_bottom
  SUBTYPE OF (blind_bottom_condition);
END_ENTITY;
```

4.5.9.2.2.2 Flat with radius hole bottom

A hole with a flat bottom which has a radius.

```
ENTITY flat_with_radius_hole_bottom
  SUBTYPE OF (blind_bottom_condition);
  corner_radius:          toleranced_length_measure;
END_ENTITY;
```

corner_radius: The radius of the corner in the bottom.

4.5.9.2.2.3 Spherical hole bottom

A hole with a spherical bottom.

```
ENTITY spherical_hole_bottom
  SUBTYPE OF (blind_bottom_condition);
  radius:          toleranced_length_measure;
END_ENTITY;
```

radius: The radius of the spherical hole bottom.

4.5.9.2.2.4 Conical hole bottom

A hole with a conical bottom, as manufactured with a standard drill. A conical bottom is a constant decrease in diameter until the diameter is 'zero', or until it becomes tangent to a tip radius.

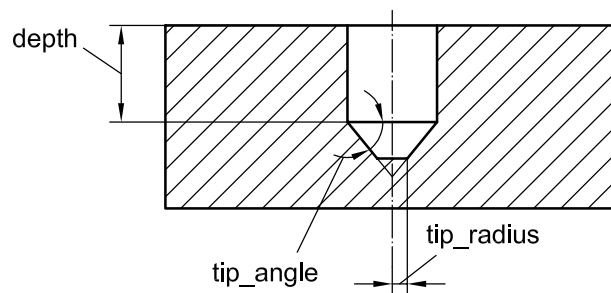


Figure 19 — Conical hole bottom

```
ENTITY conical_hole_bottom
  SUBTYPE OF (blind_bottom_condition);
  tip_angle:      plane_angle_measure;
  tip_radius:     OPTIONAL toleranced_length_measure;
END_ENTITY;
```

tip_angle: The angle of the tip.

tip_radius: Optional attribute for specification of a tip radius. Default is zero.

4.5.10 Toolpath feature

To enable the definition of tool movements not covered by the previously listed features, the toolpath_ feature has been introduced. Explicit toolpaths shall be assigned to the operations associated with this feature.

```
ENTITY toolpath_feature
  SUBTYPE OF (machining_feature);
END_ENTITY;
```

This feature shall only be used if instead of a geometry like an explicit feature, a region, etc. there is only to be described a toolpath. Regular manufacturing features should be used whenever possible. Note that **all** manufacturing features and **all** operations offer the possibility to assign manually programmed toolpaths to them. Such toolpaths will always override any automatic generation of tool movements by the machine controller and offer the CAM system complete control over the machine.

So even if some special tool movement is needed it is preferred to use a regular manufacturing feature with the associated workingsteps and attach explicit tool paths to the workingsteps.

4.5.11 Boss

A boss is a feature which has to be related to an other feature. No separate workingstep can be assigned to manufacture a boss (i.e. it is not possible to first cut the feature and then cut the boss). Instead, the boss is that material which remains unworked after the machining of a feature with a boss. For this reason, boss is not an independent machining feature but exists only as an attribute.

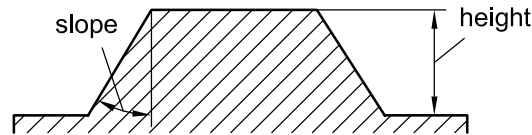


Figure 20 — Boss

```
ENTITY boss
SUBTYPE OF(machining_feature);
  its_boundary:      closed_profile;
  slope:             OPTIONAL plane_angle_measure;
(*
Informal propositions:
- The entire profile lies in the same plane.
- The profile is not self-intersecting.
*)
END_ENTITY;
```

its_boundary: The contour of the boss at the top of the boss. The contour at the bottom may be calculated by traversing the slope along the contour. Also, for the boss to have a well-defined height, its_boundary should be parallel to the bottom of the parent feature. When travelling along the profile based as defined by its sense, the material is to the left of the profile.

slope: The slope of the contour of the boss relative to the local z axis of the parent feature. The default is 0 degrees.

4.5.12 Spherical cap

This feature is circular about an axis of rotation. It consists of all points at a given distance from a point constituting its center. The center is defined by its placement, that is not located in the feature's highest point, but in the sphere's center.

```
ENTITY spherical_cap
SUBTYPE OF (machining_feature);
  internal_angle:      numeric_parameter;
  radius:              numeric_parameter;
END_ENTITY;
```

internal_angle: The size of an angle from an axis for defining a portion of a sphere.

radius: The constant distance from a point for defining a sphere.

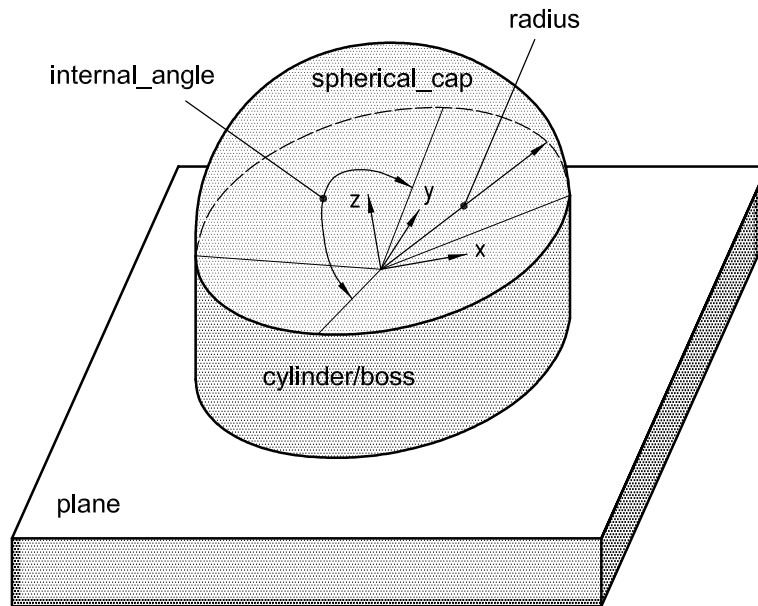


Figure 21 — Spherical cap

4.5.13 Rounded end

A rounded end is a partially circular shape passed along a linear path.

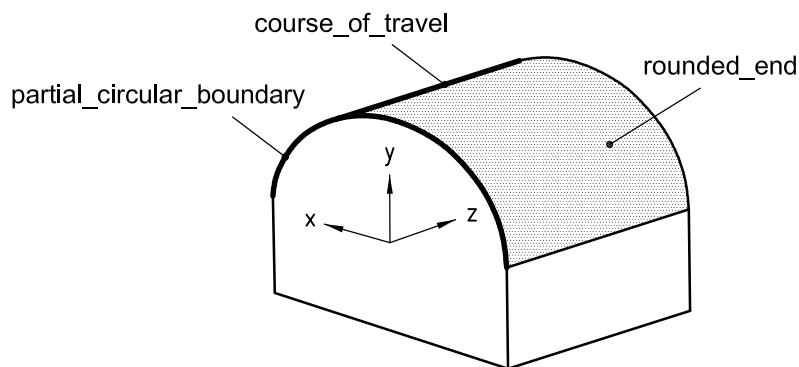


Figure 22 — Rounded end

```
ENTITY rounded_end
  SUBTYPE OF (machining_feature);
    course_of_travel: linear_path;
    partial_circular_boundary: partial_circular_profile;
END_ENTITY;
```

course_of_travel: A straight line with magnitude and direction.

partial_circular_boundary: The arc that when swept along a path defines the area on a workpiece for volume removal.

4.5.14 Compound feature

A compound feature is a feature composed of two or more features. As opposed to the replicate feature, there is no regular spacing between the elements of the compound feature. In general, they will also not be of the same type.

The volume of removed material of the compound feature is the union of all volumes of the elements of the compound feature. It is illegal to create a compound feature where the volumes of the elements are not connected in such a way that their union can be machined.

Note that the individual elements of the compound feature are associated with machining operation. In addition the compound feature includes by the attribute `its_operations` all operations of its minor elements.

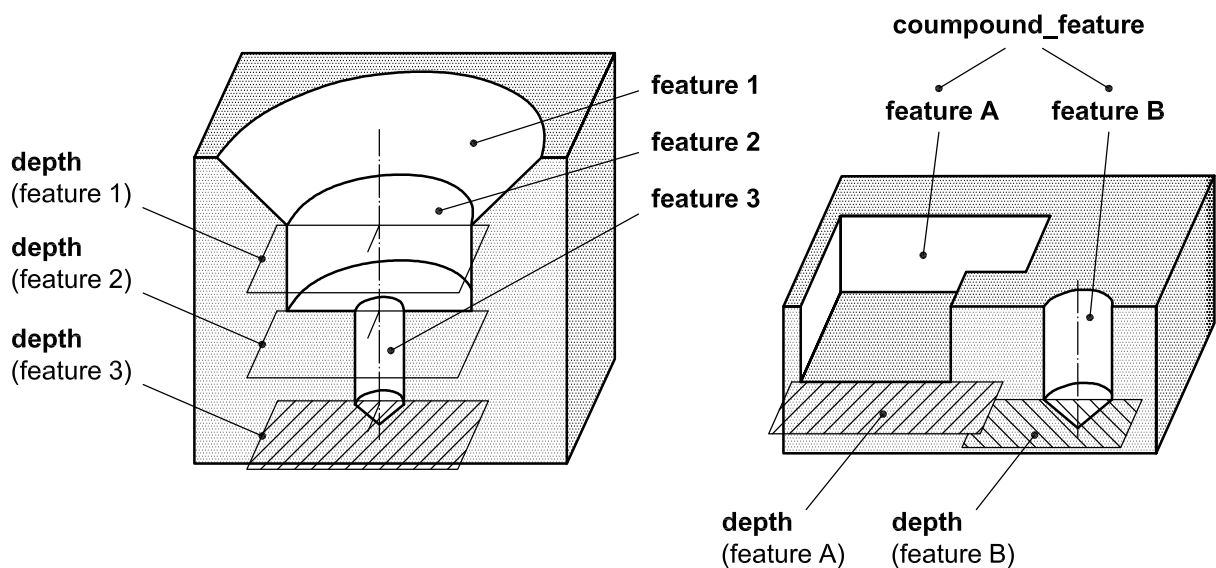


Figure 23 — Compound feature

```
ENTITY compound_feature
  SUPERTYPE OF (ONEOF(counterbore_hole, countersunk_hole))
  SUBTYPE OF (two5D_manufacturing_feature);
  elements: SET [2:?] OF compound_feature_select;
  WHERE
    WR1: SIZEOF(QUERY(e <* elements |
      SIZEOF(e\manufacturing_feature.its_operations) <> 0)) = 0;
  END_ENTITY;
```

elements: Set of features composing the compound feature. Note that the `feature_placement` attribute of the elements will specify their position relative to the local co-ordinate system of the compound feature, not relative to the workpiece co-ordinates.

4.5.14.1 Compound feature select

Selection of the elements of `compound_feature`.

```
TYPE compound_feature_select = SELECT(
  machining_feature, transition_feature
);
END_TYPE;
```

4.5.14.2 Counterbore hole

The counterbore hole is a type of compound feature that consists of two round_holes. The round hole closer to the counterbore's placement has to have a larger diameter than the one more inside of the material. The bottom of the former round_hole shall mate with the top of the latter round_hole. The orientation of counterbore_hole shall be the same as the orientation of the latter round_hole. Both round_holes shall be co-axial.

```
ENTITY counterbore_hole
  SUBTYPE OF (compound_feature);
WHERE
  WR1: SIZEOF(elements) =2;
  WR2: (SIZEOF(QUERY ( it <* SELF.elements |
    (('MACHINING_SCHEMA.ROUND_HOLE' IN TYPEOF(it))) )) = 2);
  WR3: SELF.elements[1].diameter.theoretical_size <>
    SELF.elements[2].diameter.theoretical_size;
END_ENTITY;
```

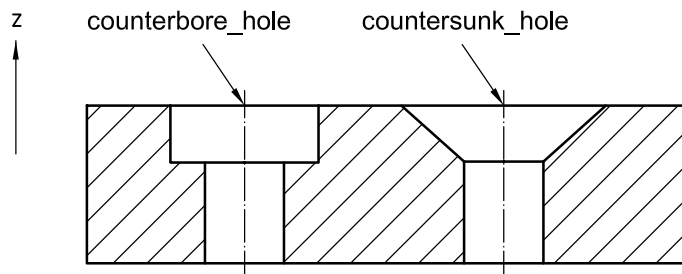


Figure 24 — Counterbore and countersunk hole

4.5.14.2.1 Countersunk hole

The countersunk hole is a type of compound feature that consists of two round_holes. The bottom of the round_hole closer to the countersunk's placement shall mate with the top of the round_hole more inside of material. The taper of the former round_hole shall be larger than the diameter of the latter round_hole, decreasing to the same diameter at the point where the two holes join. The orientation of countersunk_hole shall be the same as the orientation of the latter round_hole. Both round_holes shall be co-axial.

```
ENTITY countersunk_hole
  SUBTYPE OF (compound_feature);
WHERE
  WR1: SIZEOF(elements) =2;
  WR2: (SIZEOF(QUERY ( it <* SELF.elements |
    (('MACHINING_SCHEMA.ROUND_HOLE' IN TYPEOF(it))) )) = 2);
  WR3: SELF.elements[1].diameter.theoretical_size <>
    SELF.elements[2].diameter.theoretical_size;
  WR4: NOT EXISTS(SELF.elements[1].change_in_diameter) AND
    EXISTS(SELF.elements[2].change_in_diameter);
END_ENTITY;
```

4.5.15 Replicate feature

A replicate is an assembly of a number of similar features, e.g. a circle of holes or a mesh of holes. The feature is described only once and the number and spacing of the feature is described. Note that the attribute feature_placement of replicate_base_feature will describe the location of the feature relative to its position in the replication pattern as specified by the subtypes of replicate_feature.

```

ENTITY replicate_feature
  ABSTRACT SUPERTYPE OF (ONEOF(rectangular_pattern, circular_pattern,
    general_pattern))
  SUBTYPE OF (two5D_manufacturing_feature);
  replicate_base_feature: two5D_manufacturing_feature;
END_ENTITY;

```

replicate_base_feature: The feature which forms the basis of the replicate feature.

4.5.15.1 Circular pattern

A circular pattern of features. A complete circle of features is a special case of circular pattern.

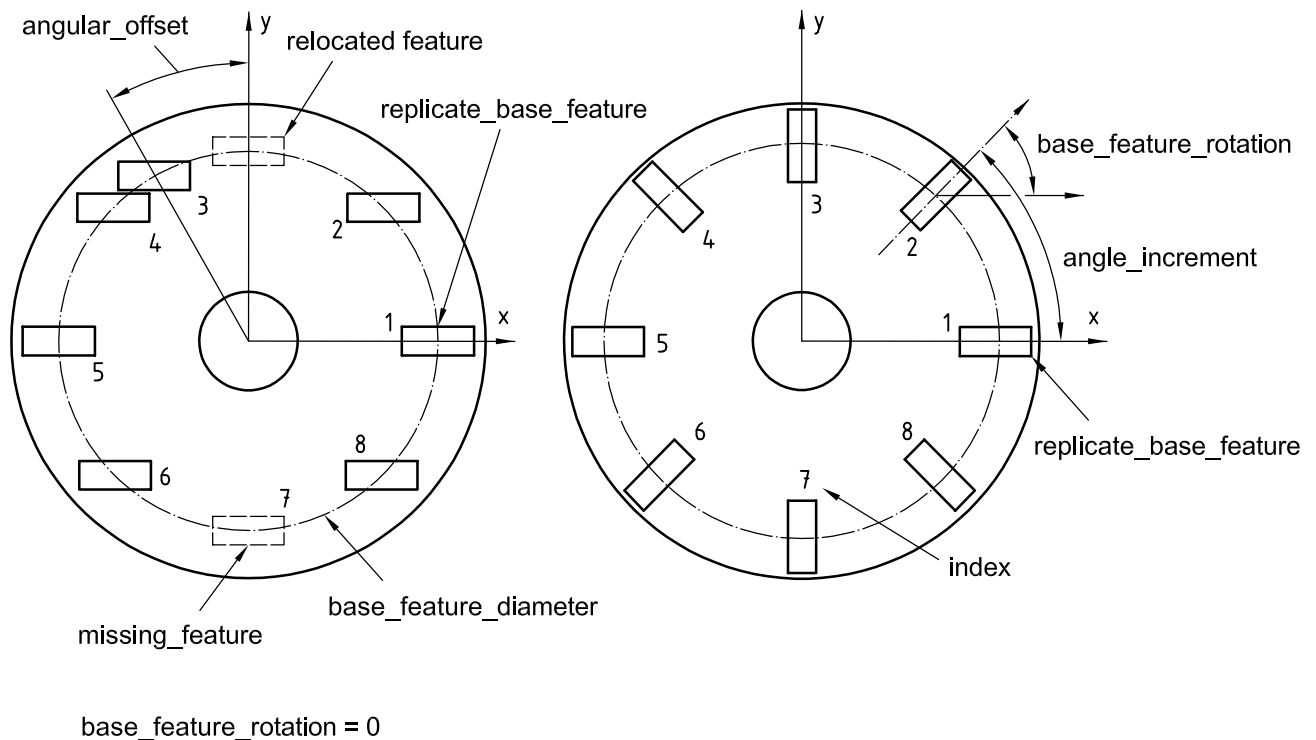


Figure 25 — Circular Pattern

```

ENTITY circular_pattern
  SUBTYPE OF (replicate_feature);
  angle_increment: plane_angle_measure;
  number_of_feature: INTEGER;
  relocated_base_feature: SET[0:?] OF circular_offset;
  missing_base_feature: SET[0:?] OF circular_omit;
  base_feature_diameter: OPTIONAL tolerated_length_measure;
  base_feature_rotation: plane_angle_measure;
END_ENTITY;

```

angle_increment: Angle between two elements of the pattern. A positive angle means a counter-clockwise increment when looking towards the plane in negative z direction.

number_of_feature: The total number of features in the replicate_feature. The maximum index of the circular pattern is calculated from number_of_feature plus the number of elements in the missing_base_feature set. The maximum index times angle_increment should not exceed 360° or the behaviour will be undefined.

relocated_base_feature:	Optional description of relocated features.
missing_base_feature:	Optional description of omitted features.
base_feature_diameter:	The diameter of the circular pattern. It has to be specified if rotate_feature is "false". If it is "true", the diameter can be calculated from the feature position and the location of the center.
base_feature_rotation	Specification of the angle to rotate one element in regard to the orientation of the previous element. The previous element is located in mathematical negative direction (counter-clockwise) to the current element.

4.5.15.1.1 Circular offset

Definition of elements offset from the circle of elements.

```
ENTITY circular_offset;
  angular_offset:      plane_angle_measure;
  index              :      INTEGER;
END_ENTITY;
```

angular_offset:	Offset of angle of the relocated element.
index:	Number of the relocated element.

4.5.15.1.2 Circular omit

Definition of elements omitted from the circle of elements.

```
ENTITY circular_omit;
  index:              INTEGER;
END_ENTITY;
```

index:	Number of the element to be omitted.
--------	--------------------------------------

4.5.15.2 Rectangular pattern

The description of elements arranged in a rectangular pattern. This may be either a grid of elements with n rows and m columns and a total number of n x m elements or a single line of m elements (n=1).

```
ENTITY rectangular_pattern
  SUBTYPE OF(replicate_feature);
  spacing:      toleranced_length_measure;
  its_direction:      direction;
  number_of_rows:      INTEGER;
  number_of_columns:      INTEGER;
  row_spacing:      OPTIONAL toleranced_length_measure;
  row_layout_direction:      OPTIONAL direction;
  relocated_base_feature:      SET[0:?] OF rectangular_offset;
  missing_base_feature:      SET[0:?] OF rectangular_omit;
  WHERE
    WR1: ( (SELF.number_of_rows > 1 )
      AND EXISTS(SELF.row_spacing)
      AND EXISTS(SELF.row_layout_direction)
    );
END_ENTITY;
```

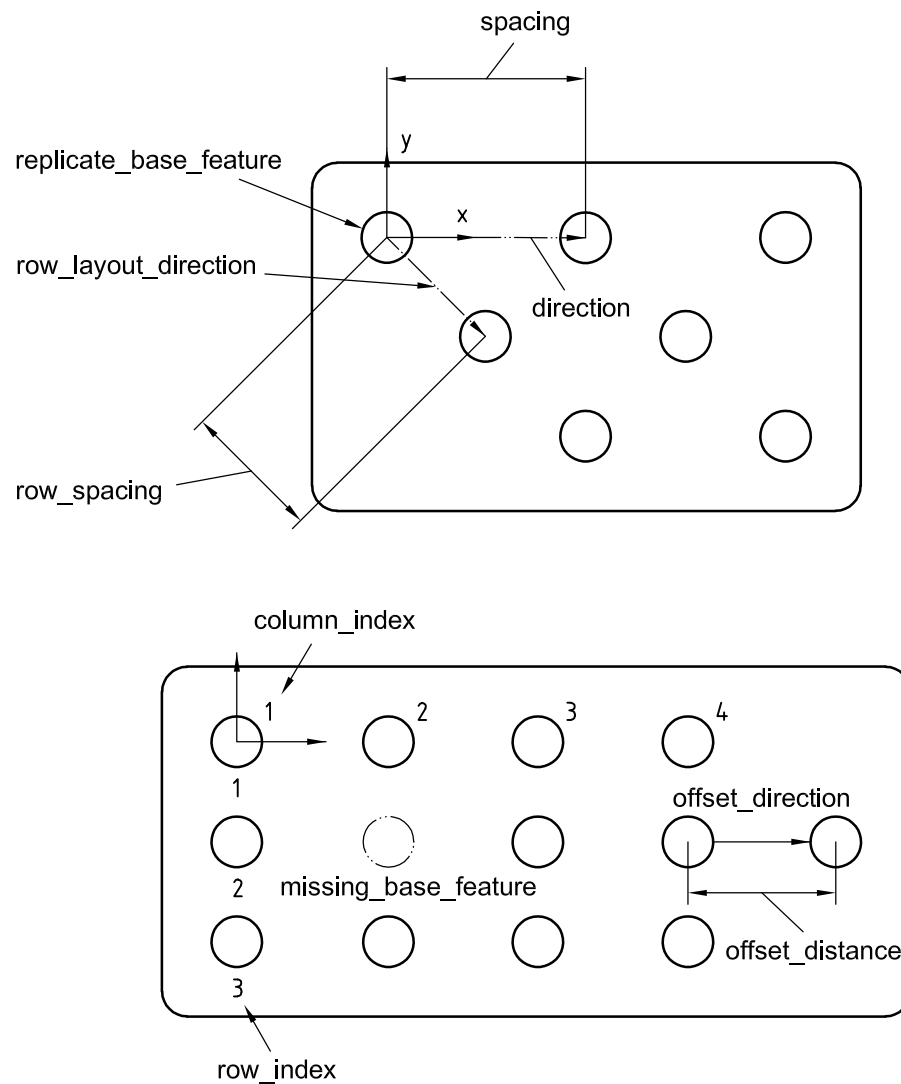


Figure 26 — Rectangular pattern

spacing:	The spacing of the columns of the pattern. If there is only one row, this is also the spacing between the elements.
its_direction:	Direction of the first row measured from the direction given in feature_placement. It points into the direction of the row, describing the order to manufacture the replicated feature.
number_of_rows:	The number of rows. Default is one, i. e. the rectangular pattern is a line of features. If number_of_rows is larger than one, the attributes row_spacing and row_layout_direction need to be specified.
number_of_columns:	The number of columns.
row_spacing:	The spacing of the rows (optional, needed if number_of_rows is larger than one).
row_layout_direction:	Optional description of the direction of the first column measured from the direction given in feature_placement. It points into the direction of the next row.
relocated_base_feature:	Optional description of relocated features.
missing_base_feature:	Optional description of missing features.

Note that the number of features can be calculated from the number_of_rows, number_of_columns, missing_base_feature, and relocated_base_feature.

4.5.15.2.1 Rectangular_offset

Entity to describe the position of a single relocated element within a rectangular pattern of elements.

```
ENTITY rectangular_offset;
  offset_direction:      direction;
  offset_distance:       length_measure;
  row_index:            INTEGER;
  column_index:         INTEGER;
END_ENTITY; -- rectangular_offset
```

offset_direction: Direction of the offset of the element.
 offset_distance: Amount of offset of the element.
 row_index: Row of the offset element.
 column_index: Column of the offset element.

4.5.15.2.2 Rectangular omit

Entity to omit one or more elements of a rectangular pattern of elements.

```
ENTITY rectangular_omit;
  row_index:            INTEGER;
  column_index:         INTEGER;
END_ENTITY; -- rectangular_omit
```

row_index: Row of the omitted element.
 column_index: Column of the omitted element.

4.5.15.3 General pattern

Definition of a general list of identical elements (used i.e. for identical holes positioned arbitrarily).

```
ENTITY general_pattern
  SUBTYPE OF (replicate_feature);
  replicate_locations:  LIST [2:?] OF axis2_placement_3d;
END_ENTITY;
```

replicate_locations: List of the placement of the features relative to the local co-ordinate system of general_pattern. The order of the features to be machined is given by the sequence of the features' placements within the list.

4.5.16 Transition feature

A transition feature may be added at the border of two features. Example: An edge round or a chamfer between two planar faces or between a planar face and a pocket. Unlike in ISO 10303-224, only those transition features which are generated using additional tool movements are considered here. Example: An edge round between a pocket and a planar face needs an additional manufacturing operation (workingstep). On the contrary, a fillet between the sides and the bottom of a pocket results from the geometry of the tool and thus needs no additional tool movement. It is therefore not considered in this context.

```
ENTITY transition_feature
  ABSTRACT SUPERTYPE OF (ONEOF(chamfer, edge_round))
  SUBTYPE OF (manufacturing_feature);
  first_feature:        machining_feature;
  second_feature:       machining_feature;
END_ENTITY;
```

first_feature: The first of the two features connected by the transition feature.
 second_feature: The second of the two features connected by the transition feature.

4.5.16.1 Chamfer

A chamfer is one of the two defined transition features for 2½D-machining. This chamfer is always an *outer* chamfer as only this can be generated in a separate manufacturing operation.

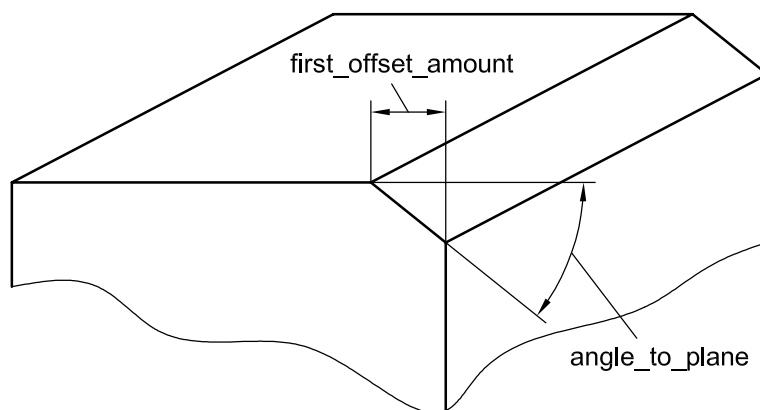


Figure 27 — Chamfer

```
ENTITY chamfer
  SUBTYPE OF (transition_feature);
  angle_to_plane: plane_angle_measure;
  first_offset_amount: toleranced_length_measure;
END_ENTITY;
```

angle_to_plane: Angle between the first feature and the chamfer face measured from the first feature.

first_offset_amount: The offset of the chamfer measure from the edge of the first feature.

4.5.16.2 Edge round (fillet)

An edge round is the other of the two defined transition features. In 2½D-machining, it is generated using a contoured tool which has to be selected according to the given fillet radius. The edge_round is always an **outer** fillet, since only this type of fillet can be manufactured in an extra manufacturing operation.

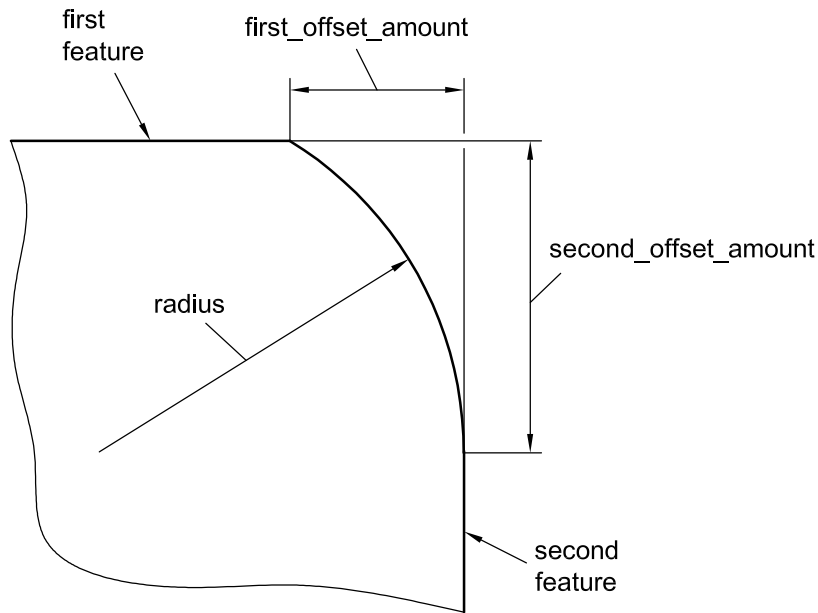


Figure 28 — Edge round

```

ENTITY edge_round
SUBTYPE OF (transition_feature);
  radius:          toleranced_length_measure;
  first_offset_amount:  OPTIONAL toleranced_length_measure;
  second_offset_amount:  OPTIONAL toleranced_length_measure;
END_ENTITY;

```

- radius:** Radius of the edge round.
- first_offset_amount:** The edge_round may or may not be tangent to the two features, this attribute specifies the location of the edge round (see figure).
- second_offset_amount:** A non-tangent edge_round may or may not be symmetrical to the two features. If it is not symmetrical, this feature specifies the location of the edge round (see figure).

4.5.17 Thread

A thread is a ridge of a uniform section on the form of a helix on the external or internal surface of a cylinder. Each thread is either a catalogue_thread or a defined_thread.

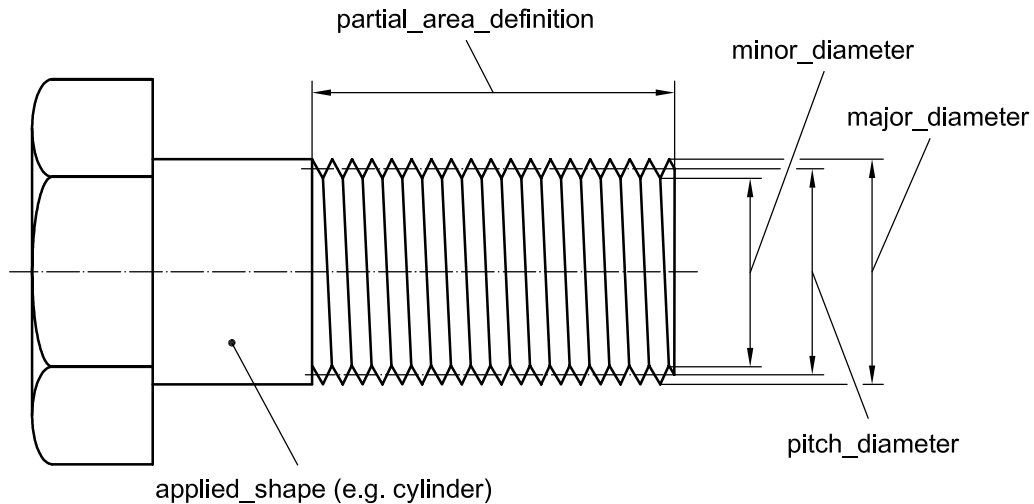


Figure 29 — Thread

```

ENTITY thread
ABSTRACT SUPERTYPE OF(ONEOF(catalogue_thread, defined_thread))
SUBTYPE OF(machining_feature);
  partial_profile:      partial_area_definition;
  applied_shape:        SET[1:?] OF machining_feature;
  inner_or_outer_thread: BOOLEAN;
  qualifier:            OPTIONAL descriptive_parameter;
  fit_class:            descriptive_parameter;
  form:                 descriptive_parameter;
  major_diameter:       length_measure;
  number_of_threads:    numeric_parameter;
  thread_hand:          descriptive_parameter;
WHERE
  WR1: ('MACHINING_SCHEMA.ROUND_HOLE' IN TYPEOF(applied_shape)) OR
        ('MACHINING_SCHEMA.CIRCULAR_CLOSED_SHAPE_PROFILE' IN
         TYPEOF(applied_shape))
        OR ('MACHINING_SCHEMA.BOSS' IN TYPEOF(applied_shape));
END_ENTITY;

```

partial_profile:	Specification of limitations of a surface to be applied on the thread.
applied_shape:	Physical shape of the workpiece that will define where the thread will be applied.
inner_or_outer_thread:	A flag specifies whether or not a thread is applied as an internal thread or an external thread.
qualifier:	Additional text information that describes a thread.
fit_class:	The value for the type of fit specification. These types are distinguished from each other by the amount of the tolerance and allowance.
form:	The definition of the shape of the thread. Various forms of threads are used to hold parts together, to adjust parts with reference to each other, or to transmit power.
major_diameter:	The dimension of the largest diameter of the thread and is applied to both an internal and external thread.
number_of_threads:	The thread pitch when used with metric unit of measure and the density of threads per inch when used with English unit of measure.
thread_hand:	A description of whether the thread is right or left handed. When viewed toward an end, a right hand winds in a clockwise direction and a left hand winds in a counterclockwise direction.

4.5.17.1 Partial area definition

A `partial_area_definition` includes the limitations of a surface for applying a thread. It places a limitation on how much and where to apply the thread on the cylindrical shape.

```
ENTITY partial_area_definition;
  effective_length:      length_measure;
  placement:            axis2_placement_3D;
  maximum_length:      OPTIONAL length_measure;
END_ENTITY;
```

effective_length The length of the thread which is usable.

placement: The location and direction of the `partial_area_definition`.

maximum_length: The dimension along a surface to apply a thread. It limits The dimensional distance limits the length along the surface axis for defining a thread.

4.5.17.2 Catalogue thread

A `catalogue_thread` is a type of thread that is defined by a document containing the information to create threads on a workpiece.

```
ENTITY catalogue_thread
  SUBTYPE OF (thread);
  documentation:      specification;
END_ENTITY;
```

documentation: Specification of the document that defines information pertaining to a thread.

4.5.17.3 Specification

A specification is a document that defines information pertaining to properties or processes for a workpiece or an aspect of a workpiece.

```
ENTITY specification;
  constraint:          SET [0:?] OF specification_usage_constraint;
  specification_id:    text;
  specification_description: OPTIONAL text;
  specification_class:  OPTIONAL text;
END_ENTITY;
```

constraint: A set of the restrictions on the specification.

specification_id: A unique identifier of the document.

specification_description: A description of the content of the specification and any notes in human interpretable prose.

specification_class: A section within a specification that is divided into classes.

4.5.17.3.1 Specification usage constraint

A `specification_usage_constraint` is a restriction on the application of information defined within a specification.

```
ENTITY specification_usage_constraint;
  element:      text;
  class_id:     text;
END_ENTITY;
```

element: The particular piece or area of information that is being restricted within the specification.

class_id: The data or range of data with respect to the element that defines the restriction imposed on the usage of the specification.

4.5.17.4 Defined thread

A defined_thread is a type of thread that is specified explicitly.

```
ENTITY defined_thread
  SUBTYPE OF (thread);
    pitch_diameter:      length_measure;
    minor_diameter:     OPTIONAL length_measure;
    crest:               OPTIONAL length_measure;
END_ENTITY;
```

pitch_diameter: The dimension of an imaginary cylinder passing through the thread so as to make equal the widths of the threads and the widths of the spaces cut by the cylinder.

minor_diameter: The dimension of the smallest diameter of the defined_thread and is applied to both an internal and external thread.

crest: The distance between the opposing points of the thread. It is formed by the intersection of the sides of the thread if extended, if necessary, beyond the top of the thread.

4.5.18 Profile

A profile is a planar outline used in the definition of a feature. A profile may be either open or closed. A profile shall be in the X-Y plane and may have an orientation that will position it in reference to the local coordinate system of a manufacturing_feature, which may require a profile as a part of its definition.

```
ENTITY profile
  ABSTRACT SUPERTYPE OF (ONEOF(closed_profile, open_profile));
    placement:          OPTIONAL axis2_placement_3d;
END_ENTITY;
```

placement: Specification where to locate the profile in reference to the orientation of the manufacturing_feature. If not given, the orientation is at the zero point of the manufacturing_feature.

4.5.18.1 Open profile

An open_profile is a type of profile that is an outline or shape with no enclosing or confining bounds. The open ends of the profile may extend infinitely.

```
ENTITY open_profile
  ABSTRACT SUPERTYPE OF
    (ONEOF (linear_profile, square_u_profile, rounded_u_profile, tee_profile,
            vee_profile, partial_circular_profile, general_profile))
  SUBTYPE OF (profile);
END_ENTITY;
```

4.5.18.1.1 Linear profile

A linear_profile is a type of open_profile that is a straight line of a specified length. The linear_profile shall have orientation parallel to the X-axis.

```
ENTITY linear_profile
  SUBTYPE OF (open_profile);
    profile_length:      numeric_parameter;
END_ENTITY;
```

profile_length: The length of the profile.

4.5.18.1.2 Square U profile

A square U profile is a type of open profile that is bounded by three lines. One is the base line and has a defined length. The other two lines begin at the ends of the base line, and extend infinitely at any obtuse or acute angle that is equal to or larger than a right angle. The corners of the square U profile need not be blended by a radius.

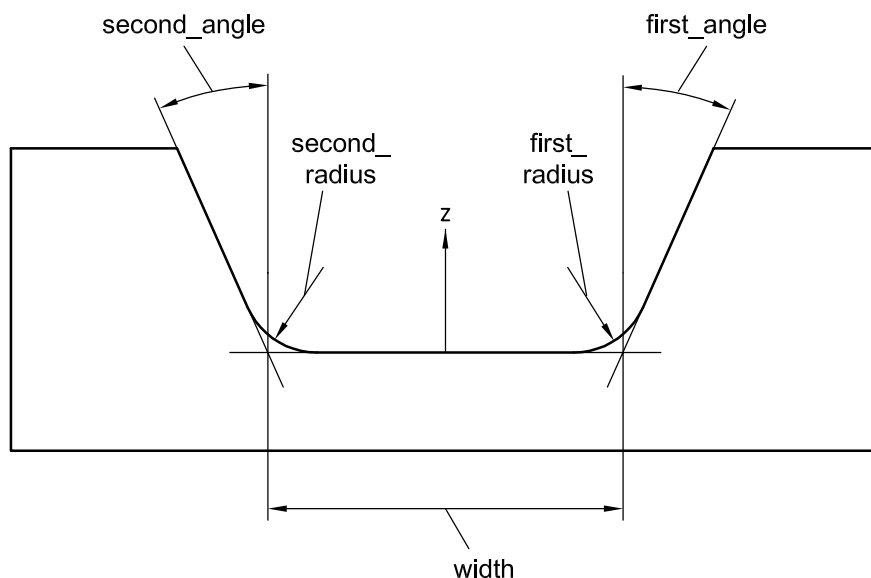


Figure 30 — Square U profile

For a rectangular slot, all values except for the width shall be zero.

```
ENTITY square_u_profile
  SUBTYPE OF (open_profile);
  width:          toleranced_length_measure;
  first_radius:   toleranced_length_measure;
  first_angle:    plane_angle_measure;
  second_radius:  toleranced_length_measure;
  second_angle:   plane_angle_measure;
END_ENTITY;
```

width:	The size of the base line for a square U profile.
first_radius:	The radius shape blend between one side of the profile and the base.
first_angle:	The angle of the one side of the profile measured against the local z axis.
second_radius:	The radius shape blend between the second side of the profile and the base.
second_angle:	The angle of the another side of the profile measured against the local z axis.

4.5.18.1.3 Rounded U profile

A rounded U profile is a type of open profile that is a shape bounded by two parallel lines and a semicircle.

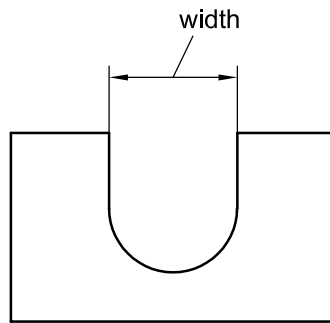


Figure 31 — Rounded U profile

```
ENTITY rounded_u_profile
  SUBTYPE OF (open_profile);
  width:          toleranced_length_measure;
END_ENTITY;
```

width: Distance across the rounded u profile.

4.5.18.1.4 T profile

A T profile is a type of open profile the cross-section of which has the shape of the letter “T”. The attributes are explained in the figure.

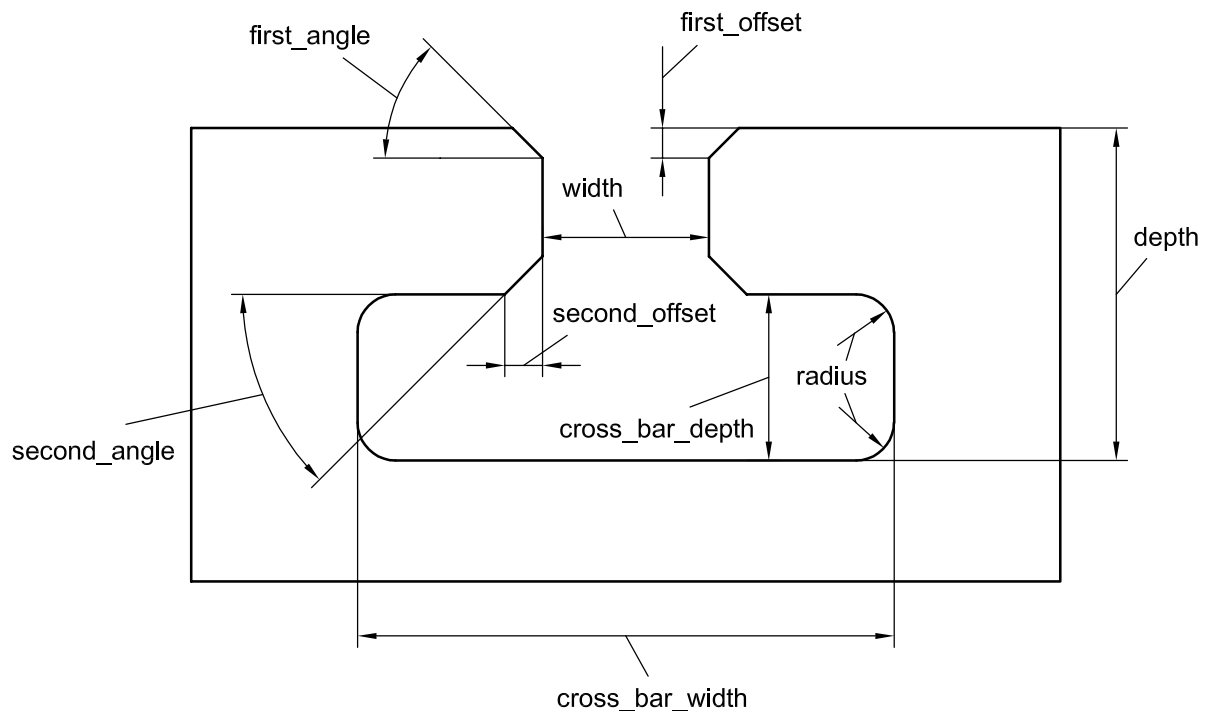


Figure 32 — T profile

```
ENTITY tee_profile
  SUBTYPE OF (open_profile);
  first_angle:      plane_angle_measure;
  second_angle:     plane_angle_measure;
  cross_bar_width:  toleranced_length_measure;
```

```

cross_bar_depth:      toleranced_length_measure;
radius:               toleranced_length_measure;
width:                toleranced_length_measure;
first_offset:         toleranced_length_measure;
second_offset:        toleranced_length_measure;
END_ENTITY;

```

first_angle: Angular measurement for creating a chamfer on the open end.

second_angle: Angular measurement for creating a chamfer between the stem and the cross bar parts of a T profile.

cross_bar_width: Width of the T cross bar.

cross_bar_depth: Depth of the T cross bar.

radius: Arc size for blending the sides of a tee cross bar.

width: Width of the T stem.

first_offset: Distance from the edge of the T stem to create a chamfer on the open end.

second_offset: Distance from the edge of the T stem to create a chamfer a distance from the edge of a surface to the finish of a chamfer.

4.5.18.1.5 V profile

A V profile is a type of open profile that is a shape bounded by two lines that connect at a point and extend infinitely. The enclosed angle is less than 180 degrees.

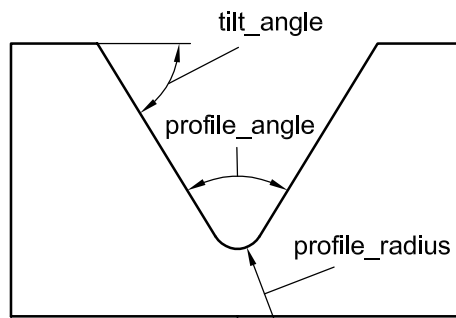


Figure 33 — V profile

```

ENTITY vee_profile
  SUBTYPE OF (open_profile);
  profile_radius:      toleranced_length_measure;
  profile_angle:       plane_angle_measure;
  tilt_angle:          plane_angle_measure;
END_ENTITY;

```

profile_radius: Size of the blend radius at the point of the V or where the two sides come together.

profile_angle: The size of the angle between the two sides of the V profile.

tilt_angle: The size of the angle between one side of the V profile and the surrounding workpiece surface.

4.5.18.1.6 Partial circular profile

A partial circular profile is a type of open_profile that is specified by an arc. The arc shall be a constant radius swept about a point. The orientation of the profile shall be positioned at the origin of the arc, with one end point of the arc on the X-axis.