INTERNATIONAL STANDARD

ISO 3592

Second edition
2000-09-15

Corrected
2001-03-15

Industrial automation systems — Numerical control of machines — NC processor output — File structure and language format

Systèmes d'automatisation industrielle — Commande numérique des machines — Informations de sortie des processeurs CN — Structure de fichier et format de langage

Reference number
ISO 3592:2000(E)

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

Printed in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 3592 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 1, *Physical device control*.

This second edition cancels and replaces the first edition (ISO 3592:1978), which has been technically revised.

Annexes A and B form a normative part of this International Standard.

Introduction

The output of a general purpose numerical control processor is information used as input to a post processor. This information is called CLDATA, which was originally derived from “cutter location data.”

CLDATA provides a general language to pass manufacturing information from a numerical control processor to a post processor, where the general language is converted to the specific format required by the particular numerical control equipment.

Numerical control is applied to many types of machines, but the language defined in this International Standard has been developed primarily for numerically controlled machine tools – hence the words “tool” and “part” are used in the description of the language to indicate the working element and processed element respectively. Many of the vocabulary words are also derived from metal working terminology.

The CLDATA reference language (RL) is stream oriented, containing special characters to delimit the elements of the RL. Annex A describes the rules for representing the RL on record oriented media, and it is this representation that is used for the purpose of describing the RL in this International Standard.

Industrial automation systems — Numerical control of machines — NC processor output — File structure and language format

1 Scope

This International Standard defines a file structure format and a language format for the representation of CLDATA on physical media.

The CLDATA reference language (RL) is used for the machining of parts. It provides for the control of technological functions and movement at the numerical control machine.

Each processor using one of the numerical control programming languages shall be capable of producing CLDATA as defined in this International Standard.

Each post processor shall be capable of using the CLDATA defined in this International Standard as input.

The RL has been developed primarily for numerically controlled machine tools.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative documents referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange.*

ISO 841:—[1)], *Numerical control of machines – Axis and motion nomenclature.*

ISO 4342:1985, *Numerical control of machines – NC processor input – Basic part program reference language.*

ISO 4343:2000, *Industrial automation systems – Numerical control of machines – NC processor output – Post processor commands.*

1) To be published. (Revision of ISO 841:1974)

3 Co-ordinate system

ISO 841 is the basis for defining the co-ordinate system of CLDATA.

The co-ordinate system is a right-handed rectangular Cartesian system, related to a part mounted on a machine and aligned with the principal linear slideways of that machine. The positive direction of movement of a component of a machine is that which causes an increasing positive dimension on the part.

In the CLDATA, the reference axes of the co-ordinate system are *x*, *y* and *z*. Co-ordinates refer to a reference point on a tool (usually the center of the tip) relative to the part co-ordinate system. CLDATA can define the following location and orientation components:

- *x* Dimension parallel to X
- *y* Dimension parallel to Y
- *z* Dimension parallel to Z
- *i* X axis component of the tool axis vector
- *j* Y axis component of the tool axis vector
- *k* Z axis component of the tool axis vector
- *l* X axis component of a secondary orientation vector
- *m* Y axis component of a secondary orientation vector
- *n* Z axis component of a secondary orientation vector

When specifying angles of planes, the positive direction is counterclockwise and the reference axis is as shown in table 1. The positive direction of angle is counterclockwise from the reference axis.

Table 1 – Reference axes

Plane	Reference axis
XY	X
YZ	Y
ZX	Z

Angles are expressed in degrees and decimal fractions of a degree.

4 General structure of CLDATA

CLDATA consists of a sequence of one or more records, which together comprise a CLDATA file.

Each record consists of a sequence of elements, to a maximum of 245, where an element is capable of representing:

a) an integer number;

b) a real number;

c) a literal character string;

d) a keyword.

Each element is composed of characters from the set of characters defined by ISO/IEC 646.

The first two elements of a record are always integers.

The remaining elements can be any combination of integer numbers, real numbers, literal character strings or keywords, respecting the syntax of the particular record.

The first element of each record contains a sequence number, commencing with 1, and incremented by 1.

The second element contains a record type code as shown in table 2.

Table 2 – CLDATA record types

Type	Name	Explanation
1 000	Original program sequence identification	This record carries the sequence and identification of the statements of the original numerical control programming language.
2 000	Integer code type post processor command	This record carries specific instructions for the post processor.
3 000	Surface data	This record carries the canonical form of the input geometry.
4 000	Relative tool position	This record carries the tool position with respect to the drive and part surfaces.
5 000	Tool position	This record carries tool position and motion vector information relating to the tool.
6 000	Post processor information	This record carries one type of information of tolerance, cutter or cut flag information.
7 000	Starting information	This record carries the tool position with respect to the startup surfaces.
8 000	Relative tool direction	This record carries information indicating tool direction with respect to the last move.
9 000	Post processor parameters	This record carries one type of multi-axis or base unit parameters.
14 000	Part program termination	This record carries the termination record.
15 000	Unsegmented tool path	This record carries unsegmented information concerning non linear tool paths.
16 000	Workpiece contour description	This record carries the workpiece contour description.
20 000	Literal type post processor command	This record carries specific instructions for the post processor.
21 000	Deferred processing command	This record is under consideration to permit user-selected input language statements to be passed to the CLDATA file in a literal form for subsequent processing.
28 000 to 32 000	Proprietary records	These records will not be standardized.

5 CLDATA file structure

5.1 General comments

5.1.1 General semantics

A CLDATA file consists of records, each in turn consisting of elements. The basic entity of CLDATA therefore is the element, which can represent either an integer number, a real number, a literal character string or a keyword.

5.1.2 Sub-contents

For

1) letters, see 5.2;
2) digits, see 5.3;
3) special characters, see 5.4;
4) characters, see 5.5;
5) symbol for a literal delimiter, see 5.6;
6) literal character strings, see 5.7;
7) unary operators, see 5.8;
8) integer numbers, see 5.9;
9) real numbers, see 5.10;
10) keywords, see 5.11;
11) symbol for an element separator, see 5.12;
12) elements, see 5.13;
13) symbol for a record separator, see 5.14;
14) records, see 5.15;
15) symbol for a file separator, see 5.16;
16) file, see 5.17.

5.1.3 Limitations

None.

5.2 Letters

5.2.1 Semantics

In general, letters have no individual meaning, being used for forming literal character strings or keywords.

5.2.2 Limitations

None.

5.2.3 Syntax

<letter> ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

5.3 Digits

5.3.1 Semantics

Digits have no individual meaning, being used for forming integer numbers, real numbers or literal character strings.

5.3.2 Limitations

None.

5.3.3 Syntax

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

5.4 Special characters

5.4.1 Semantics

Special characters are used as punctuation marks (or separators) in the CLDATA file. When special characters are used in literal character strings they are treated as characters with no syntactical significance. The special characters are

+ – unary operators, used to specify the sign of integers, reals and their exponents;

. decimal point, used to separate the whole portion of a real number from the fractional portion;

E e exponent identifier, used to adjust the value of a real number up or down by an integral power of 10;

' apostrophe, used for delimitation of a literal character string;

, comma, used as a separator between the elements of a record (<es>);

; semicolon, used as a separator between the records of a file (<rs>);

: colon, used for delimitation of a CLDATA file (<fs>).

The format control characters; horizontal tabulator (ISO/IEC 646 character code 9), line feed (code 10), form feed (code 12), carriage return (code 13) and space (code 32), have no significance except in literal character strings.

5.4.2 Limitations

None.

5.4.3 Syntax

<special_character> ::= + | – | . | E | e | ' | , | ; | :

5.5 Characters

5.5.1 Semantics

A character is a letter, digit, special character or other valid character.

5.5.2 Limitations

None.

5.5.3 Syntax

<character> ::= <letter> | <digit> | <special_character> | <other_valid_character>

Other valid characters have no significance within this International Standard but are nevertheless considered as valid input. These characters are not otherwise defined in this International Standard. They shall be manageable by the specific implementation and be selected from the character set defined by ISO/IEC 646.

5.6 Symbol for literal delimiter

5.6.1 Semantics

The apostrophe is used at the beginning and at the end of a literal character string to indicate the extent of the literal string.

5.6.2 Example

cln,5000,5,'L1',0,4.,2.5,6.;

where *cln* represents an integer value identifying the CLDATA record sequence number.

5.6.3 Limitations

None.

5.6.4 Syntax

<symbol_for_literal_delimiter> ::= '

5.7 Literal character strings

5.7.1 Semantics

A literal character string may be used in records, for listing text, or in the post processor statements for passing special information through to the post processor. The set of characters permissible is not limited to the set of letters, digits and special characters defined in this International Standard. Within a literal character string, any special characters are treated simply as characters without syntactical significance.

5.7.2 Example

cln,20000,0,PPRINT,'Set machine origin above front left corner of part';

5.7.3 Limitations

None.

5.7.4 Syntax

<literal_character_string> ::= <symbol_for_literal_delimiter> $^{0:n}$[<character>] <symbol_for_literal_delimiter>

NOTES

1) The syntax of a literal character string implies that the empty string is allowed.

2) The format control characters; horizontal tabulator (ISO/IEC 646 character code 9), line feed (code 10), form feed (code 12), carriage return (code 13) and space (code 32), are significant in literal character strings.

3) A literal character string not terminated by an apostrophe prior to an arbitrary physical record size limit (for example, column 72 in annex A) is continued from the first column on the next physical record.

4) An apostrophe is represented by two apostrophes in a non-empty literal character string.

5.8 Symbol for unary operator

5.8.1 Semantics

The special characters + and – shall be used as unary operators, in which case the operator denotes the sign of the following term.

+ denotes that the following term is to be positive;

– denotes that the following term is to be negative.

In the absence of a unary operator the following term shall be considered positive.

5.8.2 Examples

+ 45
– .82844E–3
72.6

5.8.3 Limitations

None.

5.8.4 Syntax

<unary_operator> ::= + | –

5.9 Integer numbers

5.9.1 Semantics

Integer numbers have their usual meaning, being built up of decimal digits, and optionally being preceded by a sign. The integer number is considered positive if a sign is omitted.

Leading zeros have no significance.

5.9.2 Examples

123
– 14
+ 4527

5.9.3 Limitations

There is no defined limit to the number of digits within a number, the limit being implementation dependent, although at least one digit must be specified.

5.9.4 Syntax

<integer_number> ::= $^{0:1}$[<unary_operator>] $^{1:n}$[<digit>]

5.10 Real numbers

5.10.1 Semantics

Real numbers have their usual meaning, being built up of decimal digits and a decimal character (required). Real numbers can optionally be preceded by a sign. The number is considered positive if a sign is omitted. Real numbers can also be optionally followed by an exponent raising (or lowering) the value, as indicated by the unary operator, by a integral power of 10.

Leading and trailing zeros have no significance.

5.10.2 Examples

123.
– .14
+ 45.27
.6428E–2

5.10.3 Limitations

There is no defined limit to the number of digits within a number or within the exponent, the limit being implementation dependent. No distinctions are made between whole real numbers and those containing a fractional part.

Number representation of real numbers within a computer is not necessarily exact. Therefore, approximations are used where necessary to achieve the effect of exact operations. These approximations are computer dependent.

5.10.4 Syntax

<real_number> ::= <simple_real_number> <opt_exponent>

<simple_real_number> ::= $^{0:1}$[<unary_operator>] $^{0:n}$[<digit>] **.** $^{0:n}$[<digit>]

<opt_exponent> ::= $^{0:1}$[<exponent_identifier> <integer_number>]

<exponent_identifier> ::= E|e

NOTE – The syntax of a real number implies that a solitary decimal point is allowed and denotes a real value of 0,0.

5.11 Keywords

5.11.1 Semantics

Keywords have a fixed meaning within the language. They may be regarded as entries in a vocabulary list.

5.11.2 Example

cln,20000,0,SPINDL,OFF;

5.11.3 Limitations

Keywords shall start with a letter, consist only of letters, and have a length of at least two letters.

5.11.4 Syntax

<keyword> ::= $^{2:n}$[<letter>]

Keywords shall be selected from the set of Major and Minor words defined by ISO 4343.

5.12 Symbol for element separator

5.12.1 Semantics

The comma is used as a separator between elements in a record.

5.12.2 Example

cln,1000,45,' ';

5.12.3 Limitations

None.

5.12.4 Syntax

<es> ::= ,

5.13 Elements

5.13.1 Semantics

An element is the basic entity of information in the CLDATA. It can represent either an integer number, a real number, a literal character string or a keyword.

5.13.2 Limitations

None.

5.13.3 Syntax

<element> ::= <integer_number> | <real_number> | <literal_character_string> | <keyword>

5.14 Symbol for record separator

5.14.1 Semantics

The semicolon is used as a separator between records in a file.

5.14.2 Example

cln,1000,1,' ';
cln,2000,1045,'Sample Program';
cln,1000,17,' ';
cln,6000,6,1.;

5.14.3 Limitations

None.

5.14.4 Syntax

<rs> ::= ;

NOTE – The <rs> character is the sole method of delimiting the end of record. Examples in this International Standard list records one per line to aid understanding. This does not imply that format control characters are required between records of a CLDATA file.

5.15 Records

5.15.1 Semantics

A record is a complete instruction or unit of information comparable with a sentence in a natural language.

A record consists of two or more elements, each separated from the other by an element separator character, terminated by a record termination character.

The first two elements of a record shall be type integer.

The remaining elements of a record can be any combination of integer numbers, real numbers, strings of characters or keywords, respecting the syntax of the particular record.

5.15.2 Limitations

None.

5.15.3 Syntax

<record> ::= <integer> <es> <integer> $^{0:n}$[<es> <element>] <rs>

5.16 Symbol for file separator

5.16.1 Semantics

The colon is used as a separator between files.

5.16.2 Example

cln,14000;
:

5.16.3 Limitations

None.

5.16.4 Syntax

<fs> ::= :

5.17 File

5.17.1 Semantics

A CLDATA file is a logically complete and ordered sequence of records, which after post processing will produce a correspondingly complete and ordered machine program.

A CLDATA file consists of one or more records, followed by a trailing file separator.

5.17.2 Example

```
1,20000,0,PARTNO,’EXAMPLE’;
2,6000,6,1.5,.,.75,.,.,.,3.;
3,20000,0,RAPID;
4,5000,5,’’,0,.,.,1.;
...
...
342,20000,0,END;
343,14000;
:
```

5.17.3 Limitations

None.

5.17.4 Syntax

<file> ::= $^{1:n}$[<record>] <fs>

6 Record structure

6.1 General comments

6.1.1 General semantics

Various record structures are defined in this International Standard. Each structure (or class) provides a capability to pass a specific type of information. The order and content of records within CLDATA define the type and sequence of actions desired at the numerical control machine.

6.1.2 Sub-contents

For

1) original program sequence identification record, see 6.2;
2) integer code type post processor command record, see 6.3;
3) surface data record, see 6.4;
4) relative tool position record, see 6.5;
5) tool position record, see 6.6;
6) post processor information record, see 6.7;
7) starting information record, see 6.8;
8) relative tool direction record, see 6.9;
9) post processor parameters record, see 6.10;
10) part program termination record, see 6.11;
11) unsegmented tool path record, see 6.12;
12) part contour description record, see 6.13;
13) literal type post processor command record, see 6.14;
14) deferred processing command record, see 6.15;
15) proprietary records, see 6.16.

6.1.3 Limitations

The part program termination record shall only occur as the last record of a CLDATA file.

6.1.4 Syntax

<record>::=<original_program_sequence_identification_record>|<integer_code_type_pp_command_record>|
<surface_data_record>|<relative_tool_position_record>|<tool_position_record>|<pp_information_record>|
<starting_information_record>|<relative_tool_direction_record>|<pp_parameter_record>|
<part_program_termination_record>|<unsegmented_tool_path_record>|<part_contour_description_record>|
<literal_type_pp_command_record>

6.2 Original program sequence identification

cln
1 000
original_program_sequence_number
original_program_identification
opt_original_program_statement

6.2.1 Semantics

This record carries the sequence and identification of the statements of the original numerical control programming language. This record identifies one or more subsequent CLDATA records as being produced by a particular source statement in the original part program.

cln (integer) is the CLDATA record sequence number.

1 000 (integer) identifies the record as an original program sequence identification.

original_program_sequence_number (integer) is the statement number of the original program.

original_program_identification (text) is the identification of the statement in the original program.

opt_original_program_statement (text) is an optional text element containing the original program statement.

6.2.2 Example

The following ISO 4342 statement, occurring as the 400^{th} statement in the input program, and having the text “JOB00135” in card columns 73 through 80,

FEDRAT/10,PERMIN

would be represented by the following CLDATA statements,

cln,1000,400,’JOB00135’;
cln,20000,0,FEDRAT,10.,PERMIN;

6.2.3 Limitations

None.

6.2.4 Syntax

<original_program_sequence_identification_record> ::= <integer_number> <es> 1 000
<original_program_sequence_number> <original_program_identification> <opt_original_program_statement>
<rs>

<original_program_sequence_number> ::= <es> <integer_number>

<original_program_identification> ::= <es> <literal_character_string>

<opt_original_program_statement> ::= $^{0:1}$[<es> <literal_character_string>]

6.3 Integer code type post processor command

cln
2 000
pp_command_integer_code
opt_pp_command_integer_code_par_list

The opt_pp_command_integer_code_par_list is a structure containing any combination of the following elements.

integer_number
real_number
literal_character_string
integer_code_canonical_form

The integer_code_canonical_form is an ordered structure of the following elements.

801
form_type_designator
form_size
form_name
form_subscript
form_data_list

6.3.1 Semantics

This record carries specific instructions for the post processor.

cln (integer) is the CLDATA record sequence number.

2 000 (integer) identifies the record as an integer code type post processor command.

pp_command_integer_code (integer) is the integer code of the Major word identifying the post processor command.

opt_pp_command_integer_code_par_list (various) is an optional list of post processor command parameters.

integer_number (integer) represent the integer code of the Minor words used in the post processor command.

real_number (real) represent scalar constants and scalar identifiers used in the post processor command. No distinction is made in post processor commands between integer values and real values, both are represented as real numbers in the CLDATA.

801 (integer) identifies the start of a canonical form structure.

integer_form_type_designator (integer) is a code identifying the type of the canonical form.

1 symbol for point;
2 symbol for line;
3 symbol for plane;
4 symbol for circle;
5 symbol for cylinder;
6 symbol for ellipse;
7 symbol for hyperbola;
8 symbol for cone;
9 symbol for general conic;
10 symbol for loft conic;
11 symbol for vector;
12 symbol for matrix;
13 symbol for sphere;
14 symbol for quadratic;
17 symbol for tool definition;
18 symbol for pattern;
19 symbol for space line;
20 symbol for pntvec;
21 symbol for torus.

form_size (integer) specifies the total number of elements contained within the following three entities.

form_name (text) identifies the name of the canonical form. A blank or empty string indicates an unnamed canonical form.

form_subscript (integer) identifies the subscript number of the canonical form. A value of 0 (zero) indicates an unsubscripted canonical form.

form_data_list (real) specifies the numeric values of the canonical form. See ISO 4342 for a description of the canonical forms for each form type listed in form_type_designator.

6.3.2 Example

The following ISO 4342 statement,

SPINDL/RPM,500,RANGE,2

would be represented as follows in the CLDATA,

cln,2000,1031,78,500.,145,2.;

where the integer code for SPINDL is 1031, the integer code for RPM is 78 and the integer code for RANGE is 145.

6.3.3 Limitations

The pp_command_integer_code shall be a positive, non-zero, value.

The opt_pp_command_integer_code_par_list shall contain at least one parameter if the pp_command_integer_code value exceeds or equals 1 000.

The opt_pp_command_integer_code_par_list shall be empty if the pp_command_integer_code value is in the range 1 through 999.

6.3.4 Syntax

<integer_code_type_pp_command_record> ::= <integer_number> <es> 2 000 <pp_command_integer_code>
<opt_pp_command_integer_code_par_list> <rs>

<pp_command_integer_code> ::= <es> <integer_number>

<opt_pp_command_integer_code_par_list> ::= $^{0:n}$[<es> [<integer_number> | <real_number> |
<literal_character_string> | <integer_code_canonical_form>]]

<integer_code_canonical_form> ::= 801 <es> <integer_form_type_designator> <form_size> <form_name>
<form_subscript> <form_data_list>

<integer_form_type_designator> ::= <es> <integer_number>

<form_size> ::= <es> <integer_number>

<form_name> ::= <es> <literal_character_string>

<form_subscript> ::= <es> <integer_number>

<form_data_list> ::= $^{(form\_size-2):(form\_size-2)}$[<es> <real_number>]

The pp_command_integer_code and any integer_number within the opt_pp_command_integer_code_par_list shall be selected from the set of Major and Minor word codes defined by ISO 4343.

NOTE – See 6.14 for an alternative method of carrying post processor instructions.

6.4 Surface data

cln
3 000
surface_use_designator
surface_condition_designator
surface_type_designator
surface_size
surface_name
surface_subscript
surface_canonical_list

6.4.1 Semantics

This record carries the canonical form of input geometry. This record is an indication to the post processor that the following tool position record (see 6.6 and 6.12) contains one or more co-ordinates interpolated along the specified surface type. The interpolation spacing is a function of the tolerance (see 6.7). In general practice, only circular type drive surface information (i.e. surface_type_designator 4 or 5) is output to the post processor.

cln (integer) is the CLDATA record sequence number.

3 000 (integer) identifies the record as surface data.

surface_use_designator (integer) is a code identifying the use of the surface within the NC processor.

1 symbol for part surface in a startup command;
2 symbol for drive surface;
3 symbol for check surface;
4 symbol for part surface in a continuous path command.

surface_condition_designator (integer) is a code identifying the relationship of the tool to the surface.

1 symbol for tool to the surface (TO);
2 symbol for tool past the surface (PAST);
3 symbol for tool on the surface (ON);
4 symbol for tool tangent to the surface (TANTO).

surface_type_designator (integer) is a code identifying the type of the surface.

4 symbol for circle;
5 symbol for cylinder.

surface_size (integer) specifies the total number of elements contained within the following three entities.

surface_name (text) identifies the name of the surface. A blank or empty string indicates an unnamed surface.

surface_subscript (integer) identifies the subscript number of the surface. A value of 0 (zero) indicates an unsubscripted surface.

surface_data_list (real) specifies the canonical form of the surface. See ISO 4342 for a description of the canonical forms for each the surface types listed for the surface_type_designator.

6.4.2 Example

The following ISO 4342 input statements,

GOFWD/(C1=CIRCLE/CENTER,4,2,0,RADIUS,6),PAST,L1

would have its surface data represented as follows in the CLDATA.

cln,3000,2,4,4,9,'C1',0,4.,2.,.,.,.,1.,6.;

6.4.3 Limitations

Surface_size shall be a positive integer number greater than or equal to 2.

The combined number of elements in surface_name, surface_subscript and surface_data_list shall match the value of surface_size.

6.4.4 Syntax

<surface_data_record> ::= <integer_number> <es> 3 000 <surface_use_designator> <surface_condition_designator> <surface_type_designator> <surface_size> <surface_name> <surface_subscript> <surface_data_list> <rs>

<surface_use_designator> ::= <es> <integer_number>

<surface_condition_designator> ::= <es> <integer_number>

<surface_type_designator> ::= <es> <integer_number>

<surface_size> ::= <es> <integer_number>

<surface_name> ::= <es> <literal_character_string>

<surface_subscript> ::= <es> <integer_number>

<surface_data_list> ::= $^{(\text{surface\_size-2}):(\text{surface\_size-2})}$[<es> <real_number>]

6.5 Relative tool position

cln
4 000
tool_position_designator

6.5.1 Semantics

This record carries the tool position with respect to the drive and part surfaces.

cln (integer) is the CLDATA record sequence number.

4 000 (integer) identifies the record as relative tool position data.

tool_position_designator (integer) is a code identifying the position of the tool with respect to the drive and part surfaces.

1 symbol for tool to the left of the drive surface (TLLFT);
2 symbol for tool to the right of the drive surface (TLRGT);
3 symbol for tool on the drive surface (TLON);
5 symbol for tool on the part surface (TLONPS);
6 symbol for tool offset to the part surface (TLOFPS).

The path of a tool can be controlled by two surfaces, the part surface and drive surface. In general, the part surface is the one on which the bottom of the tool is resting, and the drive surface is the one that contacts the side of the tool or that intersects the tool centre.

6.5.2 Example

The following records indicate that the tool end point is offset to the part surface and on the left side of the drive surface when viewed from behind the tool in the forward direction of motion.

cln,4000,6;
cln,4000,1;

6.5.3 Limitations

None.

6.5.4 Syntax

<relative_tool_position_record> ::= <integer_number> <es> 4 000 <tool_position_designator> <rs>

<tool_position_designator> ::= <es> <integer_number>

6.6 Tool position

cln
5 000
tool_position_type_designator
surface_name
surface_subscript
tool_position_list

6.6.1 Semantics

This record carries tool position and tool vector information relating to the tool. This information can represent the co-ordinates, orientation and surface normal of a single point or a succession of points.

cln (integer) is the CLDATA record sequence number.

5 000 (integer) identifies the record as motion data.

tool_position_type_designator (integer) is a code identifying the type of motion.

3 symbol for a startup motion (FROM);
4 symbol for a relative motion (GODLTA);
5 symbol for an absolute motion (GOTO);
6 symbol for the continuation of the previous motion data record.

surface_name (text) identifies the name of the surface being formed by the motion of the tool, or the name of the surface to which the tool is being positioned. A blank or empty string indicates an unnamed surface.

surface_subscript (integer) identifies the subscript number of the surface. A value of 0 (zero) indicates an unsubscripted surface.

tool_position_list (real) specifies one or more sets of tool position data (X,Y,Z), tool position and tool axis vector data (X,Y,Z,I,J,K), or tool position, tool axis vector and surface normal vector data (X,Y,Z,I,J,K,L,M,N).

The tool position data are the x, y, and z co-ordinates of the end point of the tool and are always expressed as an absolute dimension in cartesian co-ordinates. Therefore, a tool_position_type_designator value of 4 indicates that the absolute tool position data contained in the record was generated as a result of a relative tool position directive.

6.6.2 Example

The following ISO 4342 input statement,

GOTO/(PTA(6)=POINT/2,7,4)

would be represented by the following tool position record.

cln,5000,5,'PTA',6,2.,7.,4.;

6.6.3 Limitations

The interpretation of the tool_position_list is dependent on the most recent state of the pp_parameter_record, tool_position_format_specification subtype, tool_position_format_designator. The designator state can be one of:

0 tool position data (X,Y,Z);
1 tool position and tool axis vector data (X,Y,Z,I,J,K);
2 tool position, tool axis vector and surface normal vector data (X,Y,Z,I,J,K,L,M,N).

In the absence of this record subtype the tool_position_list is assumed to contain only X, Y and Z triplets of tool position data.

The magnitude of the I, J, K components of the tool axis vector shall equal 1,0 (one).

The magnitude of the L, M, N components of the surface normal vector shall equal 1,0 (one).

6.6.4 Syntax

<tool_position_record> ::= <integer_number> <es> 5 000 <tool_position_type_designator> <surface_name> <surface_subscript> <tool_position_list> <rs>

<tool_position_type_designator> ::= <es> <integer_number>

<surface_name> ::= <es> <literal_character_string>

<surface_subscript> ::= <es> <integer_number>

<tool_position_list> ::= $^{1:80}$[<tool_position>] | $^{1:40}$[<tool_position> <tool_axis_vector>] | $^{1:26}$[<tool_position> <tool_axis_vector> <surface_normal_vector>]

<tool_position> ::= $^{3:3}$[<es> <real_number>]

<tool_axis_vector> ::= $^{3:3}$[<es> <real_number>]

<surface_normal_vector> ::= $^{3:3}$[<es> <real_number>]

6.7 Post processor information

6.7.1 General semantics

This record carries one type of information of tolerance, tool or cut flag information.

6.7.2 Sub-contents

For

1) cut flag information, see 6.7.5;
2) tolerance information, see 6.7.6;
3) tool information, see 6.7.7.

6.7.3 Limitations

None.

6.7.4 Syntax

<pp_information_record> ::= <cut_flag_information> | <tolerance_information> | <tool_information>

6.7.5 Cut flag information

cln
6 000
1
cut_flag_designator

6.7.5.1 Semantics

This record subtype carries cut flag information. All tool positions encountered while cutting is disabled shall be ignored by the post processor. Normally this function is carried out by the NC processor and the cut flag status is for post processor information only.

cln (integer) is the CLDATA record sequence number.

6 000 (integer) identifies the record as post processor information.

1 (integer) identifies the record subtype as cut flag information.

cut_flag_designator (integer) is a code identifying the state of the cut flag.

0 Symbol for cutting enabled (CUT);
1 Symbol for cutting disabled (DNTCUT).

6.7.5.2 Example

The following ISO 4342 program segment,

```
GOTO/1,2,3
DNTCUT
GOTO/4,5,6
GOTO/7,8,9
CUT
GOTO/11,12,13
```

would result in the following CLDATA program segment, ignoring identification records for the sake of clarity.

```
cln,5000,5,'',0,1.,2.,3.;
cln,6000,1,1;
cln,6000,1,0;
cln,5000,5,'',0,7.,8.,9.;
cln,5000,5,'',0,11.,12.,13.;
```

6.7.5.3 Limitations

The post processor and NC processor shall signal an error if the cut flag is in a disabled state at program termination.

6.7.5.4 Syntax

```
<cut_flag_information>::=<integer_number><es>6 000<es>1<cut_flag_designator><rs>
<cut_flag_designator>::=<es><integer_number>
```

6.7.6 Tolerance information

cln
6 000
tolerance_designator
tolerance_specification_list

6.7.6.1 Semantics

This record subtype carries tolerance information. This information is an indication to the post processor that subsequent tool positions might deviate from the nominal co-ordinates by an amount up to and including the specified tolerance. This is not an allowance factor for the post processor, it is for for post processor information only.

cln (integer) is the CLDATA record sequence number.

6 000 (integer) identifies the record as post processor information.

tolerance_designator (integer) is a code identifying the tolerance record subtype.

4 symbol for inside (gouge) tolerance specification (INTOL);
5 symbol for outside (undercut) tolerance specification (OUTTOL).

tolerance_specification_list (real) is a list of three tolerance values specifying the amount of tolerance for the part surface, drive surface and check surface respectively.

6.7.6.2 Example

The following CLDATA records specify a total tolerance of 0,005 units, split evenly between the inside and outside methods.

cln,6000,4,2.5E–3,2.5E–3,2.5E–3;
cln,6000,5,2.5E–3,2.5E–3,2.5E–3;

6.7.6.3 Limitations

In the absence of tolerance information the post processor shall assume 0,0 inside tolerance and 0,000 5 inches of outside tolerance.

6.7.6.4 Syntax

<tolerance_information> ::= <integer_number> <es> 6 000 <tolerance_designator> <tolerance_specification_list> <rs>

<tolerance_designator> ::= <es> <integer_number>

<tolerance_specification_list> ::= $^{3:3}$[<es> <real_number>]

6.7.7 Tool information

cln
6 000
6
tool_specification_list

6.7.7.1 Semantics

This record subtype carries tool shape information. This information is an indication to the post processor that subsequent tool positions were generated based on the described tool shape. This is not a directive to the post processor to offset the tool positions in any way, it is for post processor information only.

cln (integer) is the CLDATA record sequence number.

6 000 (integer) identifies the record as post processor information.

6 (integer) identifies the record subtype as tool shape information.

tool_specification_list (real) is a list of seven values specifying various parameters of a generalized tool (see figure 1). The parameters, in order, are:

1) tool diameter, *d*;
2) corner radius, *r*;
3) horizontal distance, *e*;
4) vertical distance, *f*;
5) angle, α, between horizontal and bottom line, AB;
6) angle, β, between vertical and side line, BC;
7) tool height, *h*.

6.7.7.2 Example

The following CLDATA record specifies a tool having a diameter of 1,5 units, a corner radius of 0,5 units, and a height of 3,0 units.

cln,6000,6,1.5,.5,.25,.5,.,.,3.;

6.7.7.3 Limitations

In the absence of tool information the post processor shall assume 0,0 for each element in the tool specification list (i.e. a point tool).

6.7.7.4 Syntax

<tool_information>::=<integer_number><es>6 000<es>1<tool_specification_list><rs>

<tool_specification_list>::=$^{7:7}$[<es><real_number>]

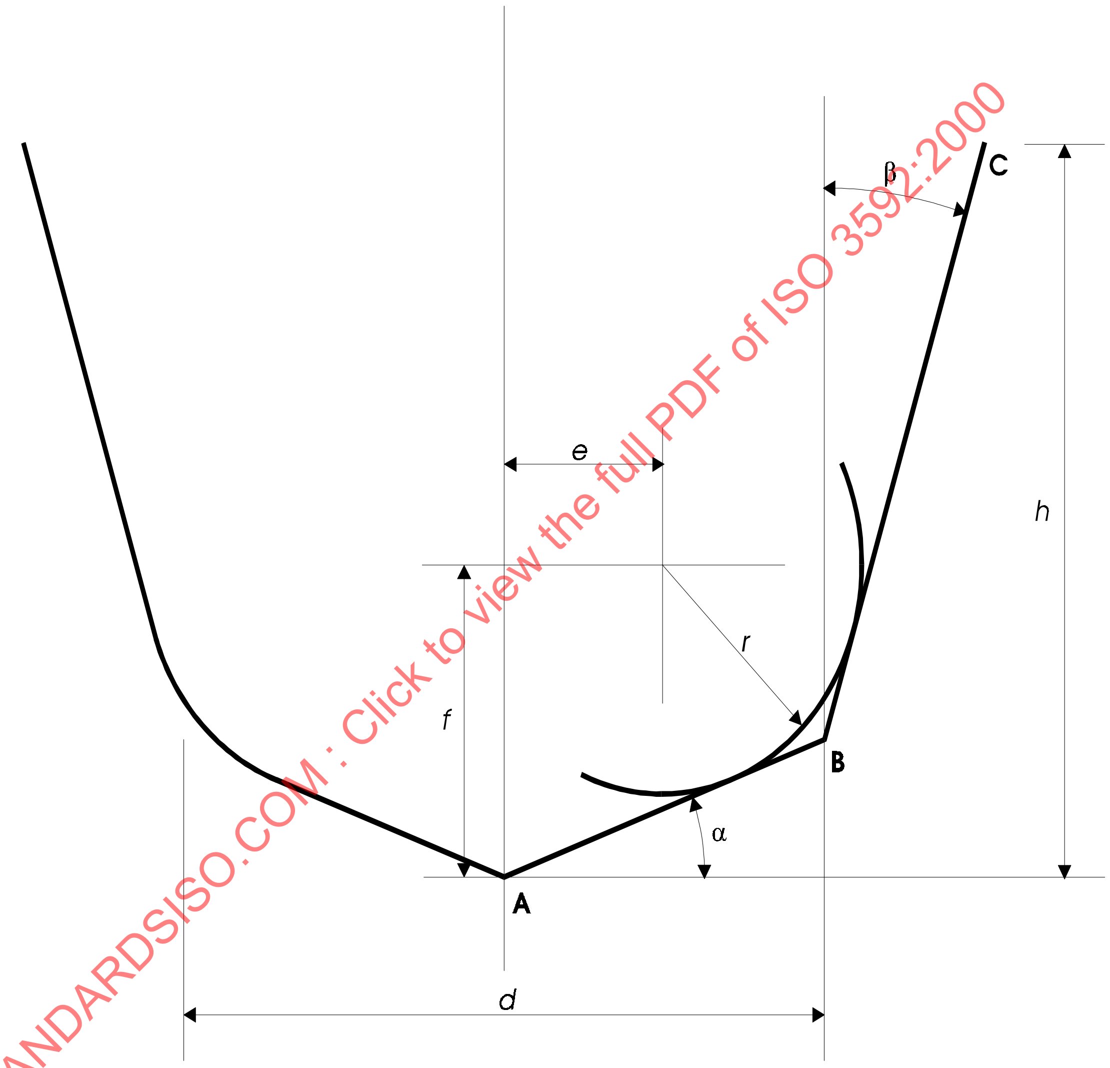

FIGURE 1 – Generalized cutter

6.8 Starting information record

cln
7 000
starting_information_designator

6.8.1 Semantics

This record carries the tool position with respect to the startup surfaces.

cln (integer) is the CLDATA record sequence number.

7 000 (integer) identifies the record as relative tool position data.

starting_information_designator (integer) is a code identifying the method in which the tool was positioned to the drive and part surfaces.

1 Symbol for direct positioning (GO);
2 Symbol for indirect positioning (OFFSET).

6.8.2 Example

The following record indicates that the tool was initially positioned using the GO command format.

cln,7000,1;

6.8.3 Limitations

None.

6.8.4 Syntax

<starting_information_record> ::= <integer_number> <es> 7 000 <starting_information_designator> <rs>

<starting_information_designator> ::= <es> <integer_number>

6.9 Relative tool direction

cln
8 000
tool_direction_designator

6.9.1 Semantics

This record carries information indicating tool direction with respect to the last move.

cln (integer) is the CLDATA record sequence number.

8 000 (integer) identifies the record as relative tool direction data.

tool_direction_designator (integer) is a code identifying the direction of the tool motion with respect to the last move.

1 symbol for motion to the left (GOLFT);
2 symbol for motion to the right (GORGT);
3 symbol for motion in the same direction (GOFWD);
4 symbol for motion in the opposite direction (GOBACK);
5 symbol for motion in the direction of the tool axis (GOUP);
6 symbol for motion in the opposite direction of the tool axis (GODOWN).

6.9.2 Example

The following record indicates that the next tool motion will be to the left in relation to the current direction of motion.

cln,8000,1;

6.9.3 Limitations

None.

6.9.4 Syntax

<relative_tool_direction_record>::=<integer_number><es>8 000<tool_direction_designator><rs>

<tool_direction_designator>::=<es><integer_number>

6.10 Post processor parameters

6.10.1 General semantics

This record carries multi-axis or units parameters.

6.10.2 Sub-contents

For

1) tool position format specification, see 6.10.5;
2) co-ordinate base units specification, see 6.10.6.

6.10.3 Limitations

None.

6.10.4 Syntax

<pp_parameter_record> ::= <tool_position_format_specification> | <coordinate_base_units_specification>

6.10.5 Tool position format specification

cln
9 000
2
format_designator

6.10.5.1 Semantics

This record subtype carries the tool position format specification.

cln (integer) is the CLDATA record sequence number.

9 000 (integer) identifies the record as post processor parameter data.

2 (integer) identifies the record subtype as tool position format specification.

format_designator (integer) is a code identifying the contents of the tool position data in the tool position record (see 6.6) The designator state can be one of:

0 tool position data only (X,Y,Z);
1 tool position and tool axis vector data (X,Y,Z,I,J,K);
2 tool position, tool axis vector and surface normal vector data (X,Y,Z,I,J,K,L,M,N).

6.10.5.2 Example

The following record indicates that subsequent tool position data contains both the tool end co-ordinates and the tool axis orientation.

cln,9000,2,1;

6.10.5.3 Limitations

None.

6.10.5.4 Syntax

<tool_position_format_specification> ::= <integer_number> <es> 9 000 <es> 2 <format_designator> <rs>

<format_designator> ::= <es> <integer_number>

6.10.6 Co-ordinate base units specification

cln
9 000
9
units_designator
opt_units_scaling_factor

6.10.6.1 Semantics

This record subtype carries the co-ordinate base units specification.

cln (integer) is the CLDATA record sequence number.

9 000 (integer) identifies the record as post processor parameter data.

9 (integer) identifies the record subtype as co-ordinate base units specification.

unit_designator (integer) is a code identifying the basic units type.

171 Symbol for millimeters;
172 Symbol for centimeters;
173 Symbol for inches;
174 Symbol for feet.

opt_unit_scaling_factor (real) is an optional amount by which to scale the basic units type. This factor allows the use of units other than millimeters, centimeters, inches and feet (for example a scaling factor of 1 000,0 and a units designator of 171 indicates that co-ordinate values are expressed in meters).

6.10.6.2 Example

The following record indicates that subsequent co-ordinate values are expressed in millimeters.

cln,9000,9,171;

6.10.6.3 Limitations

The scaling factor if specified shall be a positive, non-zero value.

In the absence of a units specification the post processor shall assume that all co-ordinates are expressed in units of inches.

6.10.6.4 Syntax

<coordinate_base_units_specification> ::= <integer_number> <es> 9 000 <es> 9 <units_designator>
<units_scaling_factor> <rs>

<units_designator> ::= <es> <integer_number>

<units_scaling_factor> ::= $^{0:1}$[<es> <real_number>]

6.11 Part program termination

cln
14 000

6.11.1 Semantics

This record carries the last record of CLDATA.

cln (integer) is the CLDATA record sequence number.

14 000 (integer) identifies the record as a part program termination record.

6.11.2 Example

The following record indicates the end of the CLDATA.

cln,14000;
:

6.11.3 Limitations

The part program termination record shall be followed by a file separator (<fs>) character.

6.11.4 Syntax

<part_program_termination_record> ::= <integer_number> <es> 14 000 <rs>

6.12 Unsegmented tool path

cln
15 000
unsegmented_tool_position_designator
surface_condition_designator
surface_type_designator
unsegmented_path_specification

6.12.1 General Semantics

This record carries unsegmented information concerning non-linear tool paths.

Non-linear motion can be represented on CLDATA in segmented or unsegmented form. The segmented form is given by the co-ordinate data sets contained in one or more tool position records (see 6.6). The unsegmented form is given by a single unsegmented tool path record which details the information to define the non-linear motion of the tool control point (for example the centre of the end of the tool for a milling machine).

Provision is made within this record to detail non-linear motions other than circular arcs (surface type designator 4 or 5) though circular arcs are the only motions at present fully described.

cln (integer) is the CLDATA record sequence number.

15 000 (integer) identifies the record as an unsegmented tool path.

unsegmented_tool_position_designator (integer) is a code identifying the relative position of the tool with respect to the surface when viewed from behind the tool in the direction of forward motion.

1 symbol for tool to the left of the surface (TLLFT);
2 symbol for tool to the right of the surface (TLRGT);
3 symbol for tool on the drive surface (TLON).

surface_condition_designator (integer) is a code identifying the relationship of the tool to the surface.

1 symbol for tool to the surface (TO);
2 symbol for tool past the surface (PAST);
3 symbol for tool on the surface (ON);
4 symbol for tool tangent to the surface (TANTO).

surface_type_designator (integer) is a code identifying the type of the surface.

4 symbol for circle;
5 symbol for cylinder.

unsegmented_path_specification (various) contains the identity of the surface, the canonical form of the surface, and path traversal information. The format of this data varies depending on the surface type.

6.12.2 Sub-contents

For

1) circular arc unsegmented path specification, see 6.12.5.

6.12.3 Limitations

None.

6.12.4 Syntax

<unsegmented_tool_path_record> ::= <integer_number> <es> 15 000 <unsegmented_tool_position_designator> <surface_condition_designator> <surface_type_designator> <unsegmented_path_specification> <rs>

<surface_condition_designator> ::= <es> <integer_number>

<surface_type_designator> ::= <es> <integer_number>

<unsegmented_path_specification> ::= <circular_arc_unsegmented_path>

6.12.5 Circular arc unsegmented path

path_specification_size
surface_name
surface_subscript
circular_canonical_list
arc_span
tool_endpoint_position

6.12.5.1 Semantics

The circular arc unsegmented path is used for surface types 4 and 5.

If the starting point is known, a circular arc motion can be defined by the co-ordinates of the circle centre, the plane of motion, the radius of the arc, a general direction and the ending point of the arc.

path_specification_size (integer) specifies the total number of elements contained within the path specification (13 for circular arcs).

surface_name (text) identifies the name of the surface. A blank or empty string indicates an unnamed surface.

surface_subscript (integer) identifies the subscript number of the surface. A value of 0 (zero) indicates an unsubscripted surface.

circular_canonical_list (real) specifies the canonical form of the surface. The seven elements of the canonical form of a circular surface, in order, are:

1) X co-ordinate of the centre of circle;
2) Y co-ordinate of the centre of circle;
3) Z co-ordinate of the centre of circle;
4) I component of axis unit vector;
5) J component of axis unit vector;
6) K component of axis unit vector;
7) radius of circle (i.e. tool path).

arc_span (real) specifies the signed angle in degrees for the circular arc path. The sign of the angle is a mathematical convention. It is based on the I, J and K components of the circle axis. A positive angle results in clockwise rotation when viewed in the direction of the circle axis vector.

tool_endpoint_position (real list) specifies the X, Y and Z co-ordinates of the endpoint of the circular arc.

6.12.5.2 Example

Figures 2 and 3 show the different record types necessary to produce segmented and unsegmented circular arc paths.

6.12.5.3 Limitations

The value of path_specification_size shall be 13.

A circular arc unsegmented path record shall be preceded by a surface data record (see 6.4) defining the surface being interpolated.

The starting position of a circular arc unsegmented path record shall be defined either by a tool position record (see 6.6) or an unsegmented tool path record (see 6.12).

6.12.5.4 Syntax

<circular_arc_unsegmented_path> ::= <path_specification_size> <surface_name> <surface_subscript>
<circular_canonical_list> <arc_span> <tool_endpoint_position>

<path_specification_size> ::= <es> <integer_number>

<surface_name> ::= <es> <literal_character_string>

<surface_subscript> ::= <es> <integer_number>

<circular_canonical_list> ::= $^{7:7}$[<es> <real_number>]

<arc_span> ::= <es> <real_number>

<tool_endpoint_position> ::= $^{3:3}$[<es> <real_number>]

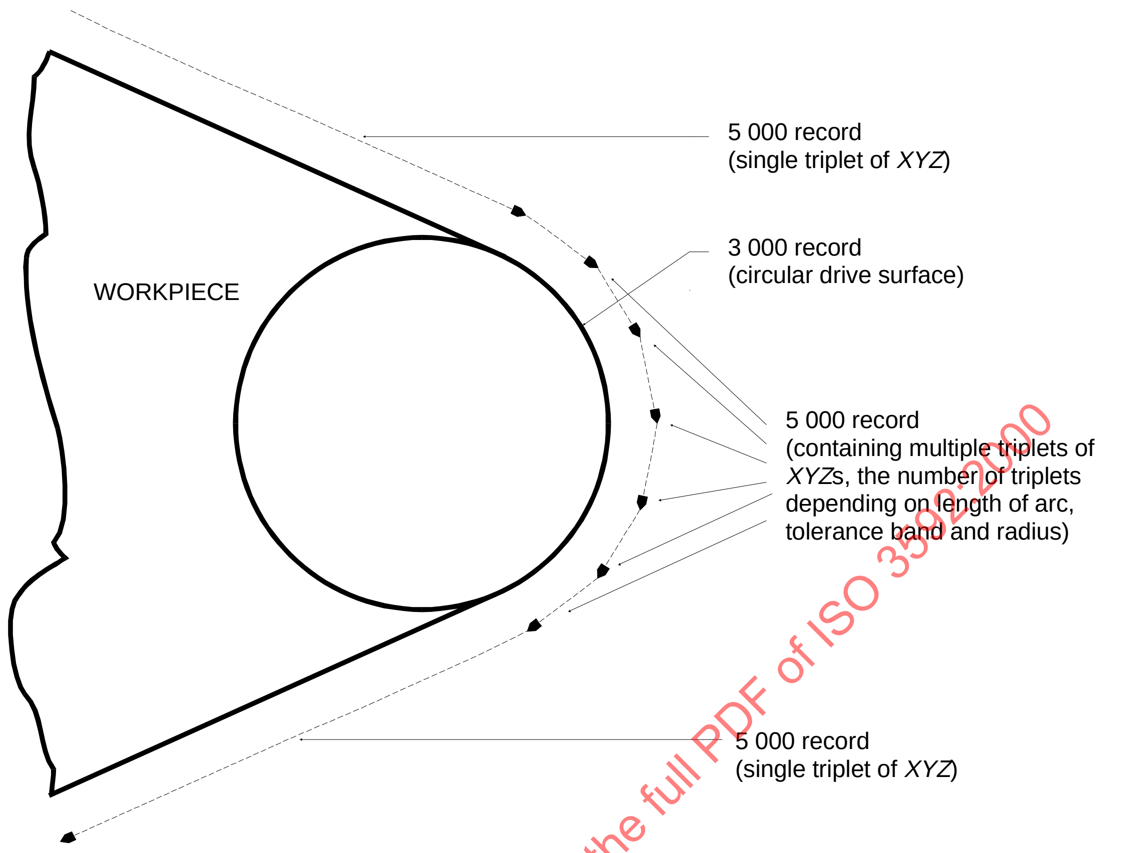

FIGURE 2 – Segmented form

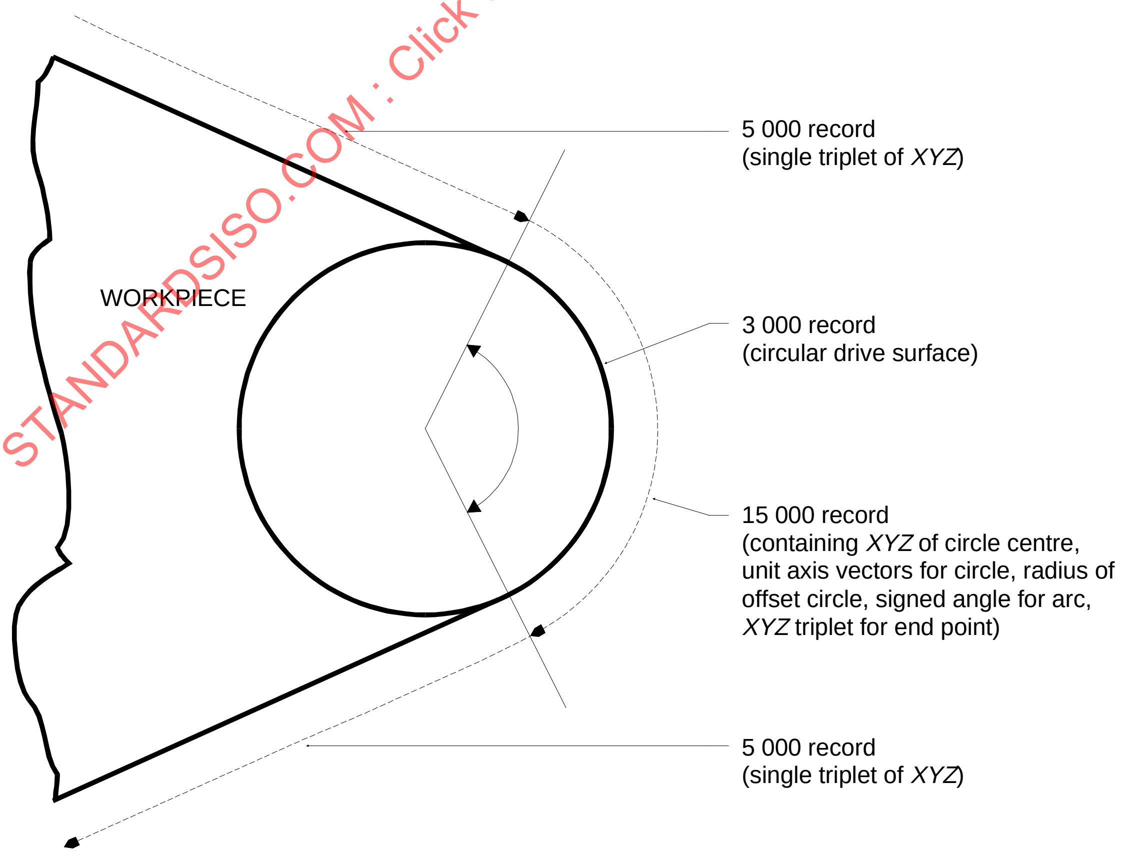

FIGURE 3 – Unsegmented form

6.13 Part contour description

6.13.1 General semantics

This record carries the part contour description. A part contour normally consists of a number of elements: straight lines, circular arcs, etceteras. A complete description of such a contour will occupy a number of part contour description records, one record being used for each element.

6.13.2 Sub-contents

For

1) rough part preparation, see 6.13.5;
2) rough part description, see 6.13.6;
3) finish part preparation, see 6.13.7;
4) finish part description, see 6.13.8.

6.13.3 Limitations

Part contour descriptions apply to two-dimensional contours only.

6.13.4 Syntax

<part_contour_description_record> ::= <rough_part_preparation> | <rough_part_description> |
<finish_part_preparation> | <finish_part_description>

6.13.5 Rough part preparation

cln
16 000
1
contour_name
contour_subscript
contour_size
contour_bounds

6.13.5.1 Semantics

The rough part preparation defines a rectangular region containing the rough part contour.

cln (integer) is the CLDATA record sequence number.

16 000 (integer) identifies the record as a part contour description.

1 (integer) identifies the record as being a rough part preparation subtype.

contour_name (text) identifies the name of the contour. A blank or empty string indicates an unnamed contour.

contour_subscript (integer) identifies the subscript number of the contour. A value of 0 (zero) indicates an unsubscripted contour.

contour_size (integer) specifies the number of contour elements in the corresponding rough part description. This number shall equal the number of rough part description records immediately following the rough part preparation record.

contour_bounds (real list) identifies the X and Y axis limits of the contour. The four boundary values, in order, are:

1) maximum X co-ordinate of part;
2) maximum Y co-ordinate of part;
3) minimum X co-ordinate of part;
4) minimum Y co-ordinate of part.

6.13.5.2 Example

The following record indicates that the rough part contour contains four elements and is bounded by an unnamed, unsubscripted region having minimum X and Y co-ordinate of –10,5 and 0,0, and maximum X and Y co-ordinates of 5,0 and 7,0.

cln,16000,1,'',0,4,5.,7.,–10.5,.;

6.13.5.3 Limitations

Contour_size shall be a positive, non-zero value matching the number of rough part description records immediately following the rough part preparation record.

6.13.5.4 Syntax

<rough_part_preparation> ::= <integer_number> <es> 16 000 <es> 1 <contour_name> <contour_subscript>
<contour_size> <contour_bounds> <rs>

<contour_name> ::= <es> <literal_character_string>

<contour_subscript> ::= <es> <integer_number>

<contour_size> ::= <es> <integer_number>

<contour_bounds> ::= $^{4:4}$[<es> <real_number>]

6.13.6 Rough part description

cln
16 000
2
contour_element_name
contour_element_subscript
contour_element_position
rough_contour_element_specification

6.13.6.1 Semantics

The rough part description defines one element in the series of elements that make up the rough part contour.

cln (integer) is the CLDATA record sequence number.

16 000 (integer) identifies the record as a part contour description.

2 (integer) identifies the record as being a rough part description subtype.

contour_element_name (text) identifies the name of the contour element. A blank or empty string indicates an unnamed contour element.

contour_element_subscript (integer) identifies the subscript number of the contour element. A value of 0 (zero) indicates an unsubscripted contour element.

contour_element_position (real list) specifies the X and Y co-ordinates of the starting point of the contour element.

rough_contour_element_specification (various) specifies the geometry of the contour element. Two element types are supported: line segments and arc segments.

The three elements (real) required for a line segment, in order, are:

1) curvature (always 0,0);
2) X component of the perpendicular vector pointing into the part;
3) Y component of the perpendicular vector pointing into the part.

The three elements (real) required for a circular arc segment, in order, are:

1) $\pm 1/r$ (curvature of circle with radius, r);
2) X co-ordinate of the centre of the circle;
3) Y co-ordinate of the centre of the circle.

The sign of the circular arc radius describes the position of the arc with reference to a chordal line joining the starting point of the arc to the ending point of the arc.

+ arc lies to the right of the line;
– arc lies to the left of the line.

6.13.6.2 Example

The following records specify a rectangular rough part description containing four linear elements, bounded by the region having minimum X and Y co-ordinate of –10,5 and 0,0, and maximum X and Y co-ordinates of 5,0 and 7,0 respectively.

cln,16000,1,’’,0,4,5.,7.,–10.5,.;
cln,16000,2,’’,0,–10.5,.,.,1.,.;
cln,16000,2,’’,0,–10.5,7.,.,.,–1.;
cln,16000,2,’’,0,5.,7.,.,–1.,.;
cln,16000,2,’’,0,5.,.,.,.,1.;

6.13.6.3 Limitations

None.