INTERNATIONAL STANDARD

ISO/IEC 7942-2

First edition
1997-09-01

Information technology — Computer graphics and image processing — Graphical Kernel System (GKS) —

Part 2: NDC metafile

Technologies de l'information — Infographie et traitement d'image — Système graphique Kernel (GKS) —

Partie 2: Métafichier NDC

Reference number
ISO/IEC 7942-2:1997(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 7942-2 was prepared by Joint Technical Committee ISO/IEC JTC1, Information technology, Subcommittee SC24, *Computer graphics and image processing.*

ISO/IEC 7942 consists of the following parts, under the general title *Information technology – Computer graphics and image processing – Graphical Kernel System (GKS) :*

Part 1: Functional description

Part 2: NDC metafile

Part 3: Audit trail

Part 4: Picture part archive

Annexes A and B form an integral part of this part of ISO/IEC 7942.

Introduction

The NDC metafile provides a file format and encodings suitable for the storage and retrieval of picture information. The file format consists of a set of elements that can be used to describe pictures in a way that is compatible between systems of different architectures and devices of differing capabilities and design. This part of ISO/IEC 7942 extends the provisions of ISO/IEC 8632:1992/Amd.2:1995.

Information technology – Computer graphics and image processing – Graphical Kernel System (GKS) – Part 2: NDC metafile

1 Scope

This part of ISO/IEC 7942 provides a file format and encodings for the storage and retrieval of GKS-94 Normalized Device Coordinate (NDC) pictures. It is an extension of the Computer Graphics Metafile, Version 4 defined by ISO/IEC 8632:1992/Amd.2: 1995 (all parts).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 7942. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 7942 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 7942-1:1994, *Information technology - Computer graphics and image processing - Graphical Kernel System (GKS) - Part 1: Functional description.*

ISO/IEC 8632:1992/Amd.2:1995, *Information technology - Computer graphics - Metafile for transfer and storage of picture description information* (all parts).

3 Definitions

For the purposes of this part of ISO/IEC 7942, the definitions given in ISO/IEC 7942-1 and ISO/IEC 8632-1/Amd.2 apply.

4 Concepts

4.1 The structure of a GKS-94 NDC metafile

The structure of a GKS-94 NDC Metafile follows the Computer Graphics Metafile (CGM) standard (ISO/IEC 8632-1:1992). It is shown in figure 1. MF is used as an abbreviation for METAFILE.

BEGIN NDC MF	MD	<picture>...	END MF

Figure 1: General form of metafile

A Computer Graphics Metafile is a collection of elements from the standardized set. The BEGIN NDC METAFILE element is followed by the METAFILE DESCRIPTOR (MD). After this the pictures follow, each logically independent of each other. Finally the Metafile is ended with an END METAFILE element.

Apart from the BEGIN NDC METAFILE, END METAFILE and Metafile Descriptor elements, the metafile is partitioned into pictures. All pictures are mutually independent. A picture consists of a BEGIN PICTURE element, a PICTURE DESCRIPTOR (PD) element, a BEGIN PICTURE BODY element, an arbitrary number of control, graphical and attribute elements and finally an END PICTURE element (see figure 2). PIC is used as an abbreviation for PICTURE and BEGIN BODY for BEGIN PICTURE BODY.

BEGIN PIC	PD	BEGIN BODY	<element>...	END PIC

Figure 2: General form of pictures

4.2 Metafile elements

Elements in CGM Version 4 are used for the mapping of NDC picture to NDC metafile wherever possible. However, some new elements are introduced to provide functionality required by the NDC metafile which cannot be described within CGM Version 4. The new CGM elements are marked by daggers (†) in the tables in this section. The scope of some CGM Version 4 elements has been extended. These elements are marked with a double dagger (‡).

4.3 Delimiter elements

The NDC metafile includes the following delimiter elements:

BEGIN NDC METAFILE†	BEGIN PICTURE BODY
END METAFILE	BEGIN APPLICATION STRUCTURE
BEGIN PICTURE	BEGIN APPLICATION STRUCTURE BODY
END PICTURE	END APPLICATION STRUCTURE

In NDC metafiles, primitives of the same type may be grouped together to form a set of primitives. The primitives to be included in the set of primitives being defined are delimited by application structure elements as described in 6.2. Application structure elements are also used in the representation of the design primitive (see 6.2.12) and the SCISSOR SET primitive attribute (see 6.3.1).

4.4 Metafile descriptor elements

4.4.1 Elements used

The NDC metafile includes the following metafile descriptor elements:

METAFILE VERSION
VDC TYPE
MAXIMUM COLOUR INDEX
METAFILE ELEMENT LIST
MAXIMUM VDC EXTENT

The METAFILE ELEMENT LIST lists at least those standardized elements that occur in the metafile.

4.4.2 Functional capability

Following the provisions of ISO/IEC 8632-1:1992/Amd.2:1995, the contents of the Computer Graphics Metafile are defined by the METAFILE ELEMENT LIST element. These shall contain a list of the non-mandatory elements that are utilized in the metafile. Several shorthand names for CGM elements are also provided for use with the METAFILE ELEMENT LIST. The NDC metafile set is designated NDC-SET. These shorthand names shall not be considered macro names, nor shall they be construed to be levels of conformance.

4.4.3 NDC-SET

The NDC-SET includes the elements which can appear in the NDC Metafile. The elements which are not marked by daggers belong to the CGM Version-4 set. The value ranges of some of the parameters of some CGM elements have been extended. These elements are marked by double daggers. The elements included in the NDC-metafile are:

<BEGIN NDC METAFILE>†
<END METAFILE>
<BEGIN PICTURE>
<BEGIN PICTURE BODY>
<END PICTURE>
<METAFILE VERSION>
<VDC TYPE>
<METAFILE ELEMENT LIST>
<MAXIMUM COLOUR INDEX>
<FONT LIST>
<CHARACTER SET LIST>
<CHARACTER CODING ANNOUNCER>
<MAXIMUM VDC EXTENT>
<VDC EXTENT>
<CLIP INDICATOR>
<CLIP RECTANGLE>
<BEGIN APPLICATION STRUCTURE>
<APPLICATION STRUCTURE ATTRIBUTES>
<BEGIN APPLICATION STRUCTURE BODY>
<END APPLICATION STRUCTURE>
<POLYLINE>
<NON-UNIFORM RATIONAL B-SPLINE>
<CONIC SECTION>†
<POLYMARKER>
<POLYGON>
<ELLIPTIC DISC>†
<TEXT>
<COLOUR SELECTION MODE>
<COLOUR VALUE EXTENT>
<COLOUR PRECISION>
<SHIELD INDICATOR>†
<GLOBAL TRANSFORMATION>†
<LOCAL TRANSFORMATION>†
<PATTERN SIZE>
<GDP>
<FILL REFERENCE POINT>
<CHARACTER HEIGHT>
<CHARACTER ORIENTATION>
<TEXT PATH>
<TEXT ALIGNMENT>
<ASPECT SOURCE FLAGS>‡
<LINE BUNDLE INDEX>
<LINE TYPE>
<LINE WIDTH>
<LINE COLOUR>
<MARKER BUNDLE INDEX>
<MARKER TYPE>
<MARKER SIZE>
<MARKER COLOUR>
<FILL BUNDLE INDEX>
<INTERIOR STYLE>
<HATCH INDEX>
<FILL COLOUR>
<EDGE VISIBILITY>
<EDGE TYPE>
<EDGE WIDTH>
<EDGE COLOUR>
<TEXT BUNDLE INDEX>
<CHARACTER EXPANSION FACTOR>
<CHARACTER SPACING>

<COLOUR MODEL>‡ <TEXT FONT INDEX>
<CELL ARRAY> <TEXT PRECISION>
<PICK IDENTIFIER> <TEXT COLOUR>
<NAMESET>† <PATTERN SIZE>
<SCISSOR IDENTIFIER>†

4.5 Control elements

In GKS-94 CLIP INDICATOR and CLIP RECTANGLE elements are the parts of the definition of an output primitive attribute, scissor set. The new control elements SCISSOR IDENTIFIER and SHIELD INDICATOR are used to provide the mapping of the GKS-94 SCISSOR SET attribute (see 4.7).

GKS elements	CGM element
CLIP INDICATOR	CLIP INDICATOR
CLIP RECTANGLE	CLIP RECTANGLE
	SCISSOR IDENTIFIER†
	SHIELD INDICATOR†

4.6 Graphical primitive elements

Graphical primitive elements are those elements that describe the visual components of a picture. Their coordinate arguments, if any, are specified in VDC units.

The mapping of GKS-94 output primitives to CGM elements is shown in the table below. An extension to CGM was needed to provide this mapping. The new CGM elements are marked by asterisks.

Each GKS-94 output primitive marked by an asterisk (*) is mapped to a set of the corresponding CGM elements delimited by BEGIN APPLICATION STRUCTURE and END APPLICATION STRUCTURE elements. The DESIGN primitive is more complicated and its mapping to the CGM elements is described in clause 6.

The CELL ARRAY primitive is mapped to a set of elements which define the way in which the colour values used in the following CELL ARRAY element are defined.

GKS elements	CGM element
SET OF POLYLINE*	POLYLINE
SET OF NURB*	NON-UNIFORM RATIONAL B-SPLINE
SET OF CONIC SECTION*	CONIC SECTION†
POLYMARKER	POLYMARKER
SET OF FILL AREA*	POLYGON
SET OF ELLIPTIC SECTOR*	CONIC SECTION†
SET OF ELLIPTIC SEGMENT*	CONIC SECTION†
SET OF ELLIPTIC DISC*	ELLIPTIC DISC†
SET OF CLOSED NURB*	NON-UNIFORM RATIONAL B-SPLINE
TEXT	TEXT
CELL ARRAY	COLOUR VALUE EXTENT
	COLOUR PRECISION
	COLOUR MODEL
	COLOUR SELECTION MODE
	CELL ARRAY
DESIGN	Several elements †
GENERALIZED DRAWING PRIMITIVE	GENERALIZED DRAWING PRIMITIVE

The CONIC SECTION element specifies a conic section defined by a 3 × 3 matrix (which specifies the conic) and two points which define the start and end points of the conic section. The last parameter of the conic

section definition is a sense flag which indicates whether it is the clockwise or anti-clockwise section that is required when the conic is closed.

The ELLIPTIC DISC element is defined by a 3 × 3 matrix which defines the ellipse.

4.7 Attribute elements

The mapping of GKS-94 output primitive attributes to CGM elements is shown in the table below. An extension to ISO/IEC 8632-1:1992/Amd.2:1995 was needed to provide this mapping. The new or revised CGM elements are marked by asterisks. The GKS-94 SCISSOR SET attribute is stored in the NDC Metafile using several elements as explained in 6.3.1.

GKS attribute	CGM elements
PICK IDENTIFIER	PICK IDENTIFIER
NAMESET	NAMESET†
SCISSOR SET	SCISSOR IDENTIFIER†
	CLIP INDICATOR
	CLIP RECTANGLE
	SHIELD INDICATOR†
GLOBAL TRANSFORMATION	GLOBAL TRANSFORMATION†
LOCAL TRANSFORMATION	LOCAL TRANSFORMATION†
PATTERN SIZE	PATTERN SIZE
PATTERN REFERENCE POINT	FILL REFERENCE POINT
TEXT HEIGHT	CHARACTER HEIGHT
TEXT UP VECTOR	CHARACTER ORIENTATION
TEXT SKEW ANGLE	CHARACTER ORIENTATION
TEXT PATH	TEXT PATH
TEXT ALIGNMENT	TEXT ALIGNMENT
CURVE INDEX	LINE BUNDLE INDEX
CURVE ASFS	ASPECT SOURCE FLAGS†
CURVE TYPE	LINE TYPE
CURVEWIDTH SCALE FACTOR	LINE WIDTH
CURVE COLOUR SPECIFIER	COLOUR SELECTION MODE
	COLOUR VALUE EXTENT
	COLOUR PRECISION
	COLOUR MODEL
	LINE COLOUR
MARKER INDEX	MARKER BUNDLE INDEX
MARKER ASFS	ASPECT SOURCE FLAGS†
MARKER TYPE	MARKER TYPE
MARKER SIZE SCALE FACTOR	MARKER SIZE
MARKER COLOUR SPECIFIER	COLOUR SELECTION MODE
	COLOUR VALUE EXTENT
	COLOUR PRECISION
	COLOUR MODEL
	MARKER COLOUR
AREA INDEX	FILL BUNDLE INDEX
AREA ASFS	ASPECT SOURCE FLAGS†
INTERIOR STYLE	INTERIOR STYLE
INTERIOR STYLE INDEX	HATCH INDEX

GKS attribute	CGM elements
INTERIOR COLOUR SPECIFIER	COLOUR SELECTION MODE COLOUR VALUE EXTENT COLOUR PRECISION COLOUR MODEL FILL COLOUR
EDGE FLAG	EDGE VISIBILITY
EDGE TYPE	EDGE TYPE
EDGEWIDTH SCALE FACTOR	EDGE WIDTH
EDGE COLOUR SPECIFIER	COLOUR SELECTION MODE COLOUR VALUE EXTENT COLOUR PRECISION COLOUR MODEL EDGE COLOUR
CHARACTER INDEX	TEXT BUNDLE INDEX
CHARACTER ASFS	ASPECT SOURCE FLAGS†
CHARACTER FONT AND PRECISION	TEXT FONT INDEX TEXT PRECISION
CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR
CHARACTER SPACING	CHARACTER SPACING
CHARACTER COLOUR SPECIFIER	COLOUR SELECTION MODE COLOUR VALUE EXTENT COLOUR PRECISION COLOUR MODEL TEXT COLOUR

4.8 Application structure elements

The NDC metafile uses application structure elements in Version 4 CGM to represent 'set of' output primitives, the design output primitive and scissor sets. The following application structure types are used.

<NDC SET OF POLYLINE>
<NDC SET OF NURB>
<NDC SET OF CONIC SECTION>
<NDC SET OF FILL AREA>
<NDC SET OF ELLIPTIC SECTOR>
<NDC SET OF ELLIPTIC SEGMENT>
<NDC SET OF ELLIPTIC DISC>
<NDC SET OF CLOSED NURB>
<NDC DESIGN PARAMETERS>
<NDC INSIDE RULE ENUMERATED>
<NDC STENCIL ATTRIBUTES>
<NDC SET OF PATHS>
<NDC CONTOUR ATTRIBUTES>
<NDC BOUNDARY SEQUENCE>
<NDC TILING>
<NDC TILING COMPONENT>
<NDC TILING COMPONENT ATTRIBUTES>
<NDC SCISSOR SET>

Application structure attributes are used to represent the parameters of design primitives, stencil, contour and tiling component attributes. The application structure attribute types used and the composition of their

associated structured data records are listed in the table below.

Attribute	Parameter	Type
<NDC DESIGN PARAMETERS>	Stencil origin	P
	Stencil transformation	(2×3)R
	Tiling origin	P
	Tiling transformation	(2×3)R
<NDC INSIDE RULE ENUMERATED>	Inside rule	(one of: evenodd, winding)E
<NDC STENCIL ATTRIBUTES>	TOPY	R
	CAPY	R
	HALFY	R
	BASEY	R
	BOTTOMY	R
	CENTREY	R
	LEFTX	R
	RIGHTX	R
	CENTREX	R
	CENTRE	P
	ORIGIN	P
	CENTRETOP	P
	CENTREBOTTOM	P
	CENTRELEFT	P
	CENTRERIGHT	P
	TOPLEFT	P
	TOPRIGHT	P
	BOTTOMLEFT	P
	BOTTOMRIGHT	P
<NDC CONTOUR ATTRIBUTES>	STYLE	(one of: solid, dashed, dotted, dashed-dotted, dashed-dashed-dotted)IX
	WIDTH	R
	CAP	(one of: butted, rounded, square)IX
	JOIN	(one of: round, bevel, mitred)IX
	mitred	R
<NDC TILING COMPONENT ATTRIBUTES>	Tiling origin	2×R
	Replication technique	(one of: dx, dy, dxdy, dydx)E
	Replication technique x value	R
	Replication technique y value	R

4.9 Metafile states

The table below shows the major metafile states in which the new metafile elements introduced in this International Standard are allowed. This table extends Table 8 in ISO/IEC 8632-1:1992/Amd.2:1995. The new elements are not allowed in any of the CGM minor states.

Element	NDC Metafile Major States								
	PCS	MDS	DR	GSS, DSS	PDS	POS	LSS	SDS	SOS
BEGIN NDC METAFILE									
SCISSOR IDENTIFIER SHIELD INDICATOR									x x
CONIC SECTION ELLIPTIC DISC				x x		x x	x x		x x
NAMESET GLOBAL TRANSFORMATION LOCAL TRANSFORMATION			x x x	x x x		x x x	x x x		

5 Abstract specification of new elements

5.1 Data type definitions and abbreviations

Data type definitions are the same as those used in CGM Version 4.

5.2 Delimiter elements

BEGIN NDC METAFILE

identifier S

This is the first element of an NDC metafile. It demarcates the beginning of the Metafile Descriptor. BEGIN NDC METAFILE shall occur exactly once in a metafile. The identifier parameter is available for use by metafile generators and interpreters in a manner that is not further standardized.

5.3 Metafile descriptor elements

COLOUR MODEL

colour model indicator (IX)

This is a revised CGM element. It may appear in the picture body. The colour model of the metafile is selected. The following values are defined:

1: RGB
2: CIELAB
3: CIELUV
4: CMYK
5: RGB - related
6: HSV
7: HLS

5.4 Control elements

SCISSOR IDENTIFIER

scissor identifier (IX)

Scissors are named by scissor identifiers.

SHIELD INDICATOR

shield indicator (one of: on, off) (E)

When the value of the shield indicator is on/off, shielding of graphical primitive elements is required or is not required.

5.5 Graphical primitive elements

CONIC SECTION

matrix33	(3 × 3) (R)
start point	(P)
end point	(P)
sense flag	(one of: clockwise, anti-clockwise) (E)

A conic section is defined by the 3 × 3 matrix, which specifies the conic and two points which define the start and end points of the conic section. The sense flag parameter indicates whether it is the clockwise or anti-clockwise section that is required when the conic is closed.

ELLIPTIC DISC

matrix33	(3 × 3) (R)

A complete ellipse is generated defined by the 3 × 3 matrix.

5.6 Attribute elements

NAMESET

nameset	(nN)

This element defines an identification attribute in the form of a set of names associated with an output primitive.

GLOBAL TRANSFORMATION

matrix23	(2 × 3) (R)

This element defines a transformation which is applied to the geometry of a primitive after the local transformation. Global transformations are used to apply a transformation to a complete object.

LOCAL TRANSFORMATION

matrix23	(2 × 3) (R)

This element defines a transformation which is applied to the geometry of a primitive to position individual output primitives.

ASPECT SOURCE FLAGS

list of: pairs of ASF type, ASF value	(one of: individual, bundled) n[E,E]

This element is revised - edge visibility ASF has been added. The designated Aspect Source Flags (ASFs) are set to the values indicated by the parameter. The following ASF types are assigned:

line type ASF	character spacing ASF
line width ASF	text colour ASF
line colour ASF	interior style ASF
marker type ASF	fill colour ASF
marker size ASF	hatch index ASF
marker colour ASF	edge type ASF
text font index ASF	edge width ASF
text precision ASF	edge colour ASF
character expansion factor ASF	edge visibility ASF†

5.7 Metafile defaults

Default values for the new attribute elements are defined as follows.

NAMESET	empty
GLOBAL TRANSFORMATION	1.0,0.0,0.0, 0.0,1.0,0.0 (the identity matrix)
LOCAL TRANSFORMATION	1.0,0.0,0.0, 0.0,1.0,0.0 (the identity matrix)

6 Mapping from NDC picture to NDC metafile

6.1 Introduction

This clause describes how an NDC picture is represented in an NDC metafile. The NDC picture consists of a strict sequence of output primitives.

The contents of the NDC picture can be stored and retrieved from metafiles. The function COPY NDC PICTURE TO NDC METAFILE will store the NDC picture The picture can be recovered at a later time and added to the current NDC picture by invoking COPY NDC METAFILE PICTURE TO NDC PICTURE.

The representation of output primitives and scissor sets makes use of the application structure functionality provided in Version 4 CGMs. In all cases STATE LIST inheritance is used to determine the graphics primitive context in which the elements are realized.

6.2 Output primitives

6.2.1 SET OF POLYLINE

The SET OF POLYLINE primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier, NDC SET OF POLYLINE, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
POLYLINE (...)
. . .
END APPLICATION STRUCTURE>

A SET OF POLYLINE primitive consisting of a single polyline can be represented by a single POLYLINE element. The listed elements generate a set of curves each of which is a sequence of connected lines defined by a point sequence.

6.2.2 SET OF NURB

The SET OF NURB primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier, NDC SET OF NURB, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
NON-UNIFORM RATIONAL B-SPLINE (...)
. . .
END APPLICATION STRUCTURE>

A SET OF NURB primitive consisting of a single NURB can be represented by a single NON-UNIFORM RATIONAL B-SPLINE element. The listed elements generate a set of curves defined by the paths each of which is a non-uniform rational B-spline.

6.2.3 SET OF CONIC SECTION

The SET OF CONIC SECTION primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC SET OF CONIC SECTION, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
CONIC SECTION (...)
. . .
END APPLICATION STRUCTURE>

A SET OF CONIC SECTION primitive consisting of a single conic section can be represented by a single CONIC SECTION element. The listed elements generate a set of curves each of which is an arc of the conic.

6.2.4 POLYMARKER

The POLYMARKER primitive is stored in the NDC Metafile as a CGM POLYMARKER element. The element generates a set of symbols of one type centered at given positions.

6.2.5 SET OF FILL AREA

The SET OF FILL AREA primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier, NDC SET OF FILL AREA,
STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
POLYGON (...)
. . .
END APPLICATION STRUCTURE>

A SET OF FILL AREA primitive consisting of a single fill area can be represented by a single POLYGON element. The listed elements generate a set of areas each of which is defined by a closed sequence of connected points.

6.2.6 SET OF ELLIPTIC SECTOR

The SET OF ELLIPTIC SECTOR primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC SET OF ELLIPTIC SECTOR, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
CONIC SECTION (...)
. . .
END APPLICATION STRUCTURE

There is no shortened form for a set of elliptic sector primitive consisting of a single elliptic sector. The listed elements generate a set of areas. Each area is closed by an elliptic arc and lines from the ends of the arc to the centre of the ellipse.

6.2.7 SET OF ELLIPTIC SEGMENT

The SET OF ELLIPTIC SEGMENT primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC SET OF ELLIPTIC SEGMENT, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
CONIC SECTION (...)
. . .
END APPLICATION STRUCTURE

There is no shortened form for a SET OF ELLIPTIC SEGMENT primitive consisting of a single elliptic segment. The listed elements generate a set of areas. Each area is closed by an elliptic arc and a line joining the ends of the arc.

6.2.8 SET OF ELLIPTIC DISC

The SET OF ELLIPTIC DISC primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

```
BEGIN APPLICATION STRUCTURE(application structure identifier,
        NDC SET OF ELLIPTIC DISC, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
ELLIPTIC DISC (...)
. . .
END APPLICATION STRUCTURE
```

A SET OF ELLIPTIC primitive consisting of a single elliptic can be represented by a single ELLIPTIC DISC element. The listed elements generate a set of areas each defined by a complete ellipse.

6.2.9 SET OF CLOSED NURB

The SET OF CLOSED NURB primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

```
BEGIN APPLICATION STRUCTURE(application structure identifier,
        NDC SET OF CLOSED NURB, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
NON-UNIFORM RATIONAL B-SPLINE (...)
. . .
END APPLICATION STRUCTURE>
```

There is no shortened form for a SET OF CLOSED NURB primitive consisting of a single closed NURB. The listed elements generate a set of areas each defined by a non-uniform rational B-spline which is closed by the first point being connected to the last.

6.2.10 TEXT

The TEXT primitive is stored in the NDC Metafile as a CGM TEXT element. The flag parameter is set to FINAL. The element generates a sequence of characters specified as a sequence of character codes relative to a specified text position.

6.2.11 CELL ARRAY

The CELL ARRAY primitive is stored in the NDC Metafile as a COLOUR SELECTION MODE element followed by a CELL ARRAY element. The COLOUR SELECTION MODE element defines whether the CELL ARRAY element contains an array of colour indices or an array of directly specified colour values. In the latter case the COLOUR SELECTION MODE element may be optionally preceeded by elements to set the colour model.

6.2.12 DESIGN PRIMITIVE

A design primitive is defined by a stencil and a tiling. Nested application structures are used to represent the design primitive and its constituent stencil and tiling. The design primitive is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC DESIGN, STATE LIST)
APPLICATION STRUCTURE ATTRIBUTE(NDC DESIGN PARAMETERS,
stencil origin, stencil transformation, tiling origin, tiling transformation)
BEGIN APPLICATION STRUCTURE BODY
stencil
tiling
END APPLICATION STRUCTURE

A *stencil* is defined in terms of a sequence of stencil outlines:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC STENCIL, STATE LIST)
APPLICATION STRUCTURE ATTRIBUTE(NDC STENCIL ATTRIBUTES, *stencil attributes*)
BEGIN APPLICATION STRUCTURE BODY
sequence of stencil outlines
END APPLICATION STRUCTURE

The *stencil attributes* are a series of ordinate and coordinate values, as described in 8.7.3 of Part 1 of this International Standard.

Stencils can be constructed in three ways: from a path contour, from a sequence of boundary definitions and by concatenation. A stencil constructed from a path contour is defined as the set of areas defined by the contours surrounding a sequence of paths. A path contour is represented as follows.

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC SET OF PATHS, STATE LIST)
APPLICATION STRUCTURE ATTRIBUTE(NDC INSIDE RULE ENUMERATED, inside rule)
APPLICATION STRUCTURE ATTRIBUTE(NDC CONTOUR ATTRIBUTES, contour attributes)
BEGIN APPLICATION STRUCTURE BODY
sequence of paths
END APPLICATION STRUCTURE

Contour attributes are described in 8.7.2 of Part 1 of this International Standard. The *sequence of paths* is any sequence of POLYLINE, NON-UNIFORM RATIONAL B-SPLINE and CONIC SECTION elements.

A sequence of boundary definitions is represented as follows.

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC BOUNDARY SEQUENCE, STATE LIST)
APPLICATION STRUCTURE ATTRIBUTE(NDC INSIDE RULE ENUMERATED, inside rule)
BEGIN APPLICATION STRUCTURE BODY
sequence of boundaries
END APPLICATION STRUCTURE

A boundary can be defined as a sequence of paths or as an area shape. A boundary is represented by POLYLINE, NON-UNIFORM RATIONAL B-SPLINE, CONIC SECTION and ELLIPTIC DISC elements. Boundaries are closed regions. If the end point of one path is not equal to the start point of the next, a single line path is added to link the two points. If the start point of the first path is not equal to the end point of the last path, then a line is added to link the two points and hence complete the closure of the boundary definition.

A tiling is represented as follows.

BEGIN APPLICATION STRUCTURE(application structure identifier, NDC TILING, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
sequence of tiling component descriptions
END APPLICATION STRUCTURE

A tiling component is represented as follows.

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC TILING COMPONENT, STATE LIST)
APPLICATION STRUCTURE ATTRIBUTE(NDC TILING COMPONENT ATTRIBUTES,
tiling origin, replication technique)
BEGIN APPLICATION STRUCTURE BODY
sequence of tile descriptions
END APPLICATION STRUCTURE

Tiles are represented in a similar way to output primitives in the CURVE, AREA and DESIGN classes. For primitives in the CURVE and AREA classes, the elements describing the geometry of the primitive are preceded by elements describing the colour of the primitive. Colour may be specified indirectly or directly. For CURVE primitives indirect colour specification is described by the elements:

COLOUR SELECTION MODE(INDEXED)
LINE COLOUR(colour index)

Direct colour specification is described by the elements:

COLOUR SELECTION MODE(DIRECT)
COLOUR VALUE EXTENT(colour mapping specifier) - This element is optional
COLOUR PRECISION(colour precision value) - This element is optional
COLOUR MODEL(positive index) - This element is optional
LINE COLOUR(colour direct)

The colour of AREA primitives is described similarly.

6.2.13 GENERALIZED DRAWING PRIMITIVE

The GENERALIZED DRAWING PRIMITIVE (GDP) primitive is stored in the NDC Metafile as a CGM GENERALIZED DRAWING PRIMITIVE element. GDP primitive provides a means of defining composite output primitives that can use the attributes of any of the primitive classes.

6.3 Output attributes

6.3.1 SCISSOR SET

Scissoring in NDC picture restricts the appearance of each output primitive either by clipping or shielding. NDC scissoring is specified by the SCISSOR SET attribute which consists of a set of named scissors. Scissors are named by scissor identifiers. A scissor consists of a clipping indicator, clipping rectangle set, shielding indicator and shielding rectangle set. The SCISSOR SET primitive attribute is stored in the NDC Metafile as a sequence of CGM elements as follows:

BEGIN APPLICATION STRUCTURE(application structure identifier,
NDC SCISSOR SET, STATE LIST)
BEGIN APPLICATION STRUCTURE BODY
SCISSOR IDENTIFIER (...)
CLIP INDICATOR (...)
CLIP RECTANGLE (...)
...
SHIELD INDICATOR (...)
CLIP RECTANGLE (...)
...
SCISSOR IDENTIFIER (...)
...
END APPLICATION STRUCTURE

Several scissors are stored between the BEGIN APPLICATION STRUCTURE BODY and END APPLICATION STRUCTURE elements. Each scissor begins by a SCISSOR IDENTIFIER element followed by CLIP INDICATOR, several CLIP RECTANGLE elements (which represent the clipping rectangle set), SHIELD INDICATOR and several CLIP RECTANGLE elements (which represent the shielding rectangle set).

6.3.2 TEXT UP VECTOR and TEXT SKEW ANGLE

TEXT UP VECTOR and TEXT SKEW ANGLE attributes are stored in the NDC Metafile as a CHARACTER ORIENTATION element. This element has four parameters which define two vectors:

x character up component
y character up component
x base component
y base component

The first is CHARACTER UP VECTOR and corresponds to the TEXT UP VECTOR in GKS-94. The second is CHARACTER BASE VECTOR which determines a skew of each character. In GKS-94, the attribute TEXT SKEW ANGLE defines a skew of each character in radians relative to the up direction (positive is anticlockwise).

TEXT UP VECTOR components correspond to the x character up component and y character up component.

If TEXT SKEW ANGLE attribute has the value α then:

x base component $-\sin \alpha$
y base component $\cos \alpha$

6.3.3 CHARACTER FONT AND PRECISION

The CHARACTER FONT AND PRECISION attribute is stored in the NDC Metafile using two elements: TEXT FONT INDEX and TEXT PRECISION.

6.3.4 COLOUR SPECIFIER

Colour may be specified indirectly or directly in GKS-94. The way of representing colour specifiers is illustrated by the curve colour specifier. For CURVE primitives indirect colour specification is described by the elements:

COLOUR SELECTION MODE(INDEXED)
LINE COLOUR(colour index)

Direct colour specification is described by the elements:

COLOUR SELECTION MODE(DIRECT)
COLOUR VALUE EXTENT(colour mapping specifier) - This element is optional
COLOUR PRECISION(colour precision value) - This element is optional
COLOUR MODEL(positive index) - This element is optional
LINE COLOUR(colour direct)

Other colour attributes are represented similarly.

7 The Character Encoding of the GKS-94 NDC Metafile

7.1 Notational conventions

The notational conventions of a character encoding of the GKS-94 NDC Metafile follows the notational conventions of a character encoding of the Computer Graphics Metafile (ISO/IEC 8632-2: 1992/Amd.2: 1995).

The GKS-94 NDC Metafile follows entering and leaving the metafile environment conventions of the Computer Graphics Metafile (ISO/IEC 8632-2: 1992/Amd.2: 1995).

7.2 Method of encoding opcodes

7.2.1 Introduction

The GKS-94 NDC Metafile follows the method of encoding opcodes of the Computer Graphics Metafile (ISO/IEC 8632-2: 1992/Amd.2: 1995).

7.2.2 Opcode assignments

Table 1 lists the opcode assignments for the CGM elements. All opcodes are from the basic opcode set. They are organized as follows: single-byte opcodes are assigned to Graphical Primitive elements except for conic, elliptical and spline output elements, where double-byte opcodes with combination 3/4 for the first byte are assigned. All other metafile elements have double-byte opcodes with the following bit combinations for the first byte:

3/0	for Delimiter elements
3/1	for Metafile Descriptor elements
3/2	for Picture Descriptor elements
3/3	for Control elements
3/5 and 3/6	for Attribute elements
3/9	for Application Structure Descriptor elements

The new CGM elements are marked by daggers (†) and changed elements by double daggers (‡) in the following table.

Opcode	7-Bit coding		8-Bit coding	
BEGIN NDC METAFILE opcode	3/0	3/7	03/0	03/7
COLOUR MODEL‡ opcode	3/1	3/3	03/1	03/3
SCISSOR IDENTIFIER† opcode	3/3	3/4	03/3	03/4
SHIELD INDICATOR† opcode	3/3	3/5	03/3	03/5
CONIC SECTION† opcode	3/4	3/2	03/4	03/2
ELLIPTIC DISC† opcode	3/4	3/3	03/4	03/3
NAMESET †opcode	3/6	3/7	03/6	03/7
GLOBAL TRANSFORMATION† opcode	3/6	3/8	03/6	03/8
LOCAL TRANSFORMATION† opcode	3/6	3/9	03/6	03/9
ASPECT SOURCE FLAGS‡opcode	3/6	3/1	03/6	03/1

Table 1 - Opcodes for metafile elements

NOTE - There is no no-op element in the character encoding. However, the character substitution mechanism allows generators to achieve this effect.

7.3 Method of encoding parameters

The GKS-94 NDC Metafile follows the method of encoding parameters of the Computer Graphics Metafile.

7.4 Representation of new elements

7.4.1 Introduction

For convenience, the 7-bit opcode is given for each element described in the remainder of this clause. To determine the 8-bit opcode, a zero is added in front of the column specification (for example, 02/1 instead of 2/1). A list of opcodes is given in table 1.

The following subclauses define the encoding of the new elements introduced for GKS-94.

7.4.2 Delimiter elements

BEGIN NDC METAFILE

<BEGIN-NDC-METAFILE-opcode: 3/0 3/7>
<string-fixed: substitution-codes>
<string-fixed: metafile-identifier>

The first parameter, substitution-code, is identical to that of the corresponding parameter of the BEGIN METAFILE element in ISO/IEC 8632-2:1992/Amd.2:1995.

7.4.3 Metafile descriptor elements

COLOUR MODEL

<COLOUR-MODEL-opcode: 3/1 3/3>
<index:colour-model>

<index: colour-model> = <integer: 1> {RGB}
| <integer: 2> {CIELAB}
| <integer: 3> {CIELUV}
| <integer: 4> {CMYK}
| <integer: 5> {RGB-related}
| <integer: 6> {HSV}
| <integer: 7> {HLS}

7.4.4 Control elements

SCISSOR IDENTIFIER

<SCISSOR-IDENTIFIER-opcode: 3/3 3/4>
<index: scissor identifier>

<index: scissor identifier>= <positive integer>

SHIELD INDICATOR

<SHIELD-INDICATOR-opcode: 3/3 3/5>
<enumerated: shield indicator>

<enumerated: shield indicator> =
<integer: 0>{off}
| <integer: 1>{on}

7.4.5 Graphical primitive elements

CONIC SECTION

<CONIC-SECTION-opcode: 3/4 3/2>
<matrix33>
<point: start point>
<point: end point>
<enumerated: sense flag >

<matrix33 > = <real: a11>
<real: a12>
<real: a13>
<real: a21>
<real: a22>
<real: a23>
<vdc: a31>
<vdc: a32>
<vdc: a33>
<enumerated: sense-flag >= <integer: 0> {clockwise}
| <integer: 1> {anti-clockwise}

ELLIPTIC DISC

<ELLIPTIC-DISC-opcode: 3/4 3/3>
<matrix33>

<matrix33> = <real: a11>
<real: a12>
<real: a13>
<real: a21>
<real: a22>
<real: a23>
<vdc: a31>
<vdc: a32>
<vdc: a33>

7.4.6 Attribute elements

NAMESET

<NAMESET-opcode: 3/6 3/7>
<name:nameset>

<name:nameset> = <integer>*

GLOBAL TRANSFORMATION

<GLOBAL-TRANSFORMATION-opcode: 3/6 3/8>
<matrix23>

<matrix23 > = <real: a11>
<real: a12>
<real: a21>
<real: a22>
<vdc: a13>
<vdc: a23>

LOCAL TRANSFORMATION

<LOCAL-TRANSFORMATION-opcode: 3/6 3/9>
<matrix23>

<matrix23 > = <real: a11>
<real: a12>
<real: a21>
<real: a22>
<vdc: a13>
<vdc: a23>

ASPECT SOURCE FLAGS

This element has been revised to include aspect source flag for EDGE VISIBILITY. It is encoded as follows:

<aspect> = <integer: 18> {edge visibility asf}

7.4.7 Application structure elements

APPLICATION STRUCTURE ATTRIBUTE

<APPLICATION-STRUCTURE-ATTRIBUTE-opcode: 3/9 2/0>
<string-fixed:application-structure-attribute-type>
<structured-data-record: data-record>

<string-fixed:application-structure-attribute-type>
= <NDC DESIGN PARAMETERS>
| <NDC INSIDE RULE ENUMERATED>
| <NDC STENCIL ATTRIBUTES>
| <NDC CONTOUR ATTRIBUTES>
| <NDC TILING COMPONENT ATTRIBUTES>

The composition of the structured data record for each type of attribute is given below.

<structured-data-record: data-record> =
<SOS>
<attribute-type-specific-record-contents>
<ST>

<attribute-type-specific-record-contents> =
| <record-contents-NDC-DESIGN-PARAMETERS>
| <record-contents-NDC-INSIDE-RULE-ENUMERATED>
| <record-contents-NDC-STENCIL-ATTRIBUTES>
| <record-contents-NDC-CONTOUR-ATTRIBUTES>
| <record-contents-NDC-TILING-COMPONENT-ATTRIBUTES>

<record-contents-NDC-DESIGN-PARAMETERS> =
<<integer: i_VDC><integer: 2><vdc: stencil origin x><vdc: stencil origin y>
<integer: i_R><integer: 4><real: sta11><real: sta12><real: sta21><real: sta22>
<integer: i_VDC><integer: 2><vdc: sta13><vdc: sta23>
<integer: i_VDC><integer: 2><vdc: tiling origin x><vdc: tiling origin y>
<integer: i_R><integer: 4><real: tla11><real: tla12><real: tla21><real: tla22>
<integer: i_VDC><integer: 2><vdc: tla13><vdc: tla23>>

<record-contents-NDC-INSIDE-RULE-ENUMERATED> =
<<integer: i_E><integer: 1><integer: 0>{evenodd}
|<integer: 1>{winding}>>

<record-contents-NDC-STENCIL-ATTRIBUTES> =
<<integer: i_R><integer: 9><real: topy>
<real: capy>
<real: halfy>
<real: basey>
<real: bottomy>
<real: centrey>
<real: leftx>
<real: rightx>
<real: centrex>
<integer: i_VDC><integer: 20><vdc: centre-x><vdc: centre-y>
<vdc: originx><vdc: originy>
<vdc: centretopx><vdc: centretopy>
<vdc: centrebottomx><vdc: centrebottomy>
<vdc: centreleftx><vdc: centrelefty>
<vdc: centrerightx><vdc: centrerighty>
<vdc: topleftx><vdc: toplefty>
<vdc: toprightx><vdc: toprighty>
<vdc: bottomleftx><vdc: bottomlefty>
<vdc: bottomrightx><vdc: bottomrighty>>

<record-contents-NDC-CONTOUR-ATTRIBUTES> =
<<integer: i_IX><integer: 1><integer: style>
{0 = solid, 1 = dashed, 2 = dotted, 3 = dashed-dotted,
4 = dashed-dashed-dotted}

<integer: i_R><integer: 1><real: width>
<integer: i_IX><integer: 1><integer: cap>
{0 = butted, 1 = rounded, 2 = square}
<integer: i_IX><integer: 1><integer: join>
{0 = round, 1 = bevel, 2 =- mitred}
<<> {for join type round, bevel}
| <<integer: i_R><integer: 1><real: mitred>>
{for join type mitred}>>

<record-contents-NDC-TILING-COMPONENT-ATTRIBUTES> =
<<integer: i_VDC><integer: 2><vdc: tiling origin x><vdc: tiling origin y>
<integer: i_E><integer: 1><<integer: 0>{DX}
| <integer: 1>{DY}
| <integer: 2>{DXDY}
| <integer: 3>{DYDX}>
< <<integer: i_R><integer: 1><real: dx value>> {if DX}
| <<integer: i_R><integer: 1><real: dy value>> {if DY}
| <<integer: i_R><integer: 2><real: dx value>
<real: dy value>> {if DXDY or DYDX}>>

8 The Binary Encoding of the GKS-94 NDC Metafile

8.1 Overall structure

The overall structure of the GKS-94 NDC Metafile as well as the general structure of the binary encoding of the GKS-94 NDC Metafile follows the Computer Graphics Metafile (CGM) standard (ISO/IEC 8632-3:1992/Amd.2:1995).

8.2 Primitive data forms

The GKS-94 NDC Metafile follows the primitive data forms of the Computer Graphics Metafile (ISO/IEC 8632-3: 1992/Amd.2: 1995).

8.3 Representation of abstract parameter types

The GKS-94 NDC Metafile follows the representation of abstract parameter types of the Computer Graphics Metafile (ISO/IEC 8632-3: 1992/Amd.2: 1995).

8.4 Representation of each element

8.4.1 Introduction

The GKS-94 NDC Metafile follows the method of presentation of the Computer Graphics Metafile (ISO/IEC 8632-3: 1992/Amd.2: 1995).

The elements are grouped according to their class:

0 Delimiter elements
1 Metafile Descriptor elements
2 Picture Descriptor elements
3 Control elements
4 Graphical Primitive elements
5 Attribute elements
9 Application Structure Descriptor elements

The following subclauses define the encoding of the new elements introduced for GKS-94.

8.4.2 Delimiter elements

Element Class 0	Element Id	Parameter Type	Parameter List Length	Parameter Range
BEGIN NDC METAFILE	24	SF	BS	SR

Code Description
1 BEGIN NDC METAFILE: has 1 parameter:
P1: (string fixed) metafile name

8.4.3 Metafile descriptor elements

The set of colour models supported has been extended to include all the models supported by GKS-94.

Element Class 1	Element Id	Parameter Type	Parameter List Length	Parameter Range
COLOUR MODEL	19	IX	BIX	+IXR

Code Description
19 COLOUR MODEL: has 1 parameter:
P1:(index) colour model: valid values are
1 RGB
2 CIELAB
3 CIELV
4 CMYK
5 RGB-related
6 HSV
7 HLS

8.4.4 Control elements

Element Class 3	Element Id	Parameter Type	Parameter List Length	Parameter Range
SCISSOR IDENTIFIER	21	IX	BIX	+IXR
SHIELD INDICATOR	22	E	BE	{0,1}

Code Description
21 SCISSOR IDENTIFIER: has 1 parameter:
P1: (index) scissor identifier

22 SHIELD INDICATOR: has 1 parameter:
P1: (enumerated) shield indicator: valid values are
0 off
1 on

8.4.5 Graphical primitive elements

Element Class 4	Element Id	Parameter Type	Parameter List Length	Parameter Range
CONIC SECTION	30	9R 2P, E,	9BR+ 2BP+ BE	RR VDCR {0,1}
ELLIPTIC DISC	31	9R	9BR	RR

Code Description
30 CONIC SECTION: has 4 parameters:
P1:The next 9 values are components of a transformation matrix consisting of:
a11:(real) component
a12:(real) component
a13:(real) component
a21:(real) component
a22:(real) component
a23:(real) component
a31:(real) component

a32:(real) component
a33:(real) component
P2:(point)start point
P3:(point)end point
P4: (enumerated) sense flag
0 clockwise
1 anti-clockwise

31 ELLIPTIC DISC: has 1 parameter:
P1:The next 9 values are components of a transformation matrix consisting of:
a11:(real) component
a12:(real) component
a13:(real) component
a21:(real) component
a22:(real) component
a23:(real) component
a31:(real) component
a32:(real) component
a33:(real) component

8.4.6 Attribute elements

Element Class 5	Element Id	Parameter Type	Parameter List Length	Parameter Range
ASPECT SOURCE FLAGS	35	n(E,E)	n(2BE)	{0..18} {0,1}
NAMESET	52	nN	nBN	NR
GLOBAL TRANSFORMATION	53	6R	6BR	RR
LOCAL TRANSFORMATION	54	6R	6BR	RR

Code Description
35 ASPECT SOURCE FLAGS: has up to 19 parameter-pairs.
This element has been revised to include an aspect source flag for EDGE VISIBILITY.
It is encoded as follows:
18 edge visibility ASF

52 NAMESET: has a variable parameter list:
P1-Pn: (name) n names

53 GLOBAL TRANSFORMATION: has 1 parameter:
P1: The next 6 values are components of a 2 × 3 transformation matrix.
a11:(real)
a12:(real)
a21:(real)
a22:(real)
a13:(real)
a23:(real)

54 LOCAL TRANSFORMATION: has 1 parameter:
P1: The next 6 values are components of a 2 × 3 transformation matrix.
a11:(real)
a12:(real)

a21:(real)
a22:(real)
a13:(real)
a23:(real)

8.4.7 Structure attribute elements

Structure attribute elements are a specialization of the APPLICATION STRUCTURE ATTRIBUTE element defined in ISO/IEC 8632:1992/Amd.2:1995.

Element Class 9	Element Id	Parameter Type	Parameter List Length	Parameter Range
APPLICATION STRUCTURE ATTRIBUTE	1	SF, SDR	BS+BS	SR, SR

Code Description

1 APPLICATION STRUCTURE ATTRIBUTE: has 2 paramters.

P1: (string fixed) external attribute name: valid values are

NDC DESIGN PARAMETERS
NDC INSIDE RULE ENUMERATED
NDC STENCIL ATTRIBUTES
NDC CONTOUR ATTRIBUTES
NDC TILING COMPONENT ATTRIBUTES

P2: (structured data record) attribute parameter record, each record contains one or more members, each of which is comprised of [data type indicator, data element count, data element(s)]. Valid values of the records are:

[(integer: i_VDC)(integer: 2)(vdc: stencil-origin-x)(vdc: stencil-origin-y)
(integer: i_R)(integer: 4)(real: sta11)(real: sta12)(real: sta21)(real: sta22>
(integer: i_VDC)(integer: 2)(vdc: sta13)(vdc: sta23>
(integer: i_VDC)(integer: 2)(vdc: tiling-origin-x)(vdc: tiling-origin-y>
(integer: i_R)(integer: 4)(real: tla11)(real: tla12)(real: tla21)(real: tla22>
(integer: i_VDC)(integer: 2)(vdc: tla13)(vdc: tla23)], for
NDC DESIGN PARAMETERS
[(integer: i_E)(integer: 1)(integer: inside-rule)
{0 = evenodd, 1 = winding}], for NDC INSIDE RULE ENUMERATED
[(integer: i_R)(integer: 9)(real: topy)
(real: capy)
(real: halfy)
(real: basey)
(real: bottomy)
(real: centrey)
(real: leftx)
(real: rightx)
(real: centrex)
(integer: i_VDC)(integer: 20)(vdc: centre-x)(vdc: centre-y)
(vdc: origin-x)(vdc: origin-y)
(vdc: centretop-x)(vdc: centretop-y)
(vdc: centrebottom-x)(vdc: centrebottom-y)
(vdc: centeleft-x)(vdc: centreleft-y)
(vdc: centreright-x)(vdc: centreright-y)

(vdc: topleft-x)(vdc: topleft-y)
(vdc: topright-x)(vdc: topright-y)
(vdc: bottomleft-x)(vdc: bottomleft-y)
(vdc: bottomright-x)(vdc: bottomright-y)], for NDC STENCIL ATTRIBUTES
[(integer: i_IX)(integer: 1)(integer: style)
{0 = solid, 1 = dashed, 2 = dotted, 3 = dashed-dotted, 4 = dashed-dashed-dotted}
(integer: i_R)(integer: 1)(real: width)
(integer: i_IX)(integer: 1)(integer: cap))
{0 = butted, 1 = rounded, 2 = square}]
(integer: i_IX)(integer: 1)(integer: join)
{0 = round, 1 = bevel, 2 = mitred}
[[null_SDR] {for join type round, bevel}
| [(integer: i_R)(integer: 1)(real: mitred)]
{for join type mitred}], for NDC CONTOUR ATTRIBUTES
[(integer: i_VDC)(integer: 2)(vdc: tiling-origin-x)
(vdc: tiling-origin-y)
(integer: i_E)(integer: 1)(integer: replication-technique)
{0 = DX, 1 = DY, 2 = DXDY, 3 = DYDX}
[[(integer: i_R)(integer: 1)(real: dx_value)] {if DX}
| [(integer: i_R)(integer: 1)(real: dy_value)] {if DY}
| [(integer: i_R)(integer: 2)(real: dx_value)
(real: dy_value)] {if DXDY or DYDX}]], for
NDC TILING COMPONENT ATTRIBUTES

9 The Clear Text Encoding of the GKS-94 NDC Metafile

9.1 Notational conventions

The notational conventions of a clear text encoding of the GKS-94 NDC Metafile follow the notational conventions of a cler text encoding of the Computer Graphics Metafile (ISO/IEC 8632-4:1992/Amd.2:1995).

9.2 Encoding parameter types

Th derived type DIRECTCOLR is expanded to include the specification of colours in the HSV and HLS colour models.

```
<H S V>          ::= <I:H><SEP><I:S><SEP><I:V>
<H L S>          ::= <I:H><SEP><I:L><SEP><I:S>

<DIRECTCOLR>     ::= <RED GREEN BLUE> {if COLOUR MODEL is RGB}
                   | <L A B> {if COLOUR MODEL is CIELAB}
                   | <L U V> {if COLOUR MODEL is CIELUV}
                   | <CYAN MAGENTA YELLOW BLACK>
                       {if COLOUR MODEL is CMYK}
                   | <A B C> {if COLOUR MODEL is RGB-related}
                   | <H S V> {if COLOUR MODEL is HSV}
                   | <H L S> {if COLOUR MODEL is HLS}
```

9.3 Forming names

9.3.1 Introduction

The approach to forming words used to name elements and enumeration types follows the approach taken by the Clear Text Encoding of the Computer Graphics Metafile with the following amendments.

9.3.2 Words deleted

COMPONENT

9.3.3 Words used unabbreviated

ANTI-CLOCKWISE	ENUMERATED	RULE
CLOCKWISE	EVENODD	SCISSOR
CONIC	GLOBAL	SECTION
CONTOUR	INSIDE	STENCIL
DESIGN	LOCAL	TILING
DISC	NAMESET	WINDING
ELLIPTIC	NDC	

9.3.4 Abbreviations

ELLIPTIC	ELLIP
PARAMETERS	PARAM

9.3.5 The derived names of new elements

Metafile Element	Element Name
BEGIN NDC METAFILE	BEGNDCMF
CONIC SECTION	CONICSECTION
ELLIPTIC DISC	ELLIPDISC
NAMESET	NAMESET
SCISSOR IDENTIFIER	SCISSORID
SHIELD INDICATOR	SHIELD
GLOBAL TRANSFORMATION	GLOBALTRAN
LOCAL TRANSFORMATION	LOCALTRAN

9.4 Encoding the NDC Metafile elements

9.4.1 Introduction

The GKS-94 NDC Metafile follows the method of encoding metafile elements of the Computer Graphics Metafile (ISO/IEC 8632-4:1992/Amd.2:1995).

9.4.2 Encoding delimiter elements

BEGIN NDC METAFILE ::= BEGNDCMF
<OPTSEP>
<SF:NAME>
<TERM>

9.4.3 Encoding metafile descriptor elements

COLOUR MODEL ::= COLRMODEL
<SOFTSEP>
<I:MODELINDEX>
{1=RGB, 2=CIELAB,
3=CIELUV, 4=CMYK, 5=RGB-related,
6=HSV, 7=HLS}
<TERM>

9.4.4 Encoding control elements

SCISSOR IDENTIFIER ::= SCISSORID
<SOFTSEP>
<I:SCISSORID> {positive}
<TERM>

SHIELD INDICATOR ::= SHIELD
<SOFTSEP>
<OFFION>
<TERM>

9.4.5 Encoding graphical primitive elements

CONIC SECTION ::= CONICSECTION
<SOFTSEP>
<MATRIX33:MATRIX33>
<SEP>
<P:POINT>
<SEP>

```
                                    <P:POINT>
                                    <SEP>
                                    <CLOCKWISE | ANTI-CLOCKWISE>
                                  <TERM>
<MATRIX33:MATRIX33>          ::=  <<R:a11><SEP>
                                    <R:a12><SEP>
                                    <R:a13><SEP>
                                    <R:a21><SEP>
                                    <R:a22><SEP>
                                    <R:a23><SEP>
                                    <V:a31><SEP>
                                    <V:a32><SEP>
                                    <V:a33><SEP>>>
ELLIPTIC DISC                ::=  ELLIPDISC
                                    <SOFTSEP>
                                    <MATRIX33:MATRIX33>
                                  <TERM>
```

9.4.6 Encoding attribute elements

```
NAMESET                      ::=  NAMESET
                                    <<SOFTSEP>
                                    <I:NAME><<SEP><I:NAME>>*>o
                                  <TERM>
GLOBAL TRANSFORMATION        ::=  GLOBALTRAN
                                    <SOFTSEP>
                                    <TM: MATRIX23>
                                  <TERM>
LOCAL TRANSFORMATION         ::=  LOCALTRAN
                                    <SOFTSEP>
                                    <TM: MATRIX23>
                                  <TERM>
```

ASPECT SOURCE FLAGS

This element has been revised to include aspect source flag for EDGE VISIBILITY. It is encoded by the ASFNAME value EDGEVIS. The production for ASFNAME is extended accordingly.

The pseudo-ASF ALLEDGE is extended to include EDGEVIS. The meaning of ALLEDGE is therefore:

ALLEDGE: set EDGETYPE, EDGEVIS, EDGEWIDTH and EDGECOLR as indicated.

9.4.7 Encoding of application structure elements

The composition of the structured data record for each type is given below.

```
<SDR:NDCDESIGNPARAM>         ::=  <<I:i_VDC><SEP><I:2><SEP><VDC:STORIGX>
                                    <SEP><VDC:STORIGY>
                                  <SEP><I:i_R><SEP><I:4><SEP><R:STA11><SEP><R:STA12>
                                    <SEP><R:STA21><SEP><R:STA22>
                                  <SEP><I:i_VDC><SEP><I:2>
                                    <SEP><VDC:STA13><SEP><VDC:STA23>
                                  <SEP><I:i_VDC><SEP><I:2>
                                    <SEP><VDC:TLORIGX><SEP><VDC:TLORIGY>
                                  <SEP><I:i_R><SEP><I:4><SEP><R:TLA11><SEP><R:TLA12>
                                    <SEP><R:TLA21><SEP><R:TLA22>
```

<SEP><I:i_VDC><SEP><I:2>
<SEP><VDC:TLA13><SEP><VDC:TLA23>>

<SDR:NDCINSIDERULEENUMERATED> ::= <<I:i_E><SEP><I:1><SEP><EVENODD|WINDING>>

<SDR:NDCSTENCILATTR> ::= <<I:i_R><SEP><I:9><SEP><R:TOPY>
<SEP><R:CAPY>
<SEP><R:HALFY>
<SEP><R:BASEY>
<SEP><R:BOTTOMY>
<SEP><R:CENTREY>
<SEP><R:LEFTX>
<SEP><R:RIGHTX>
<SEP><R:CENTREX>
<SEP><I:i_VDC><SEP><I:20>
<SEP><VDC:CENTREX>
<SEP><VDC:CENTREY>
<SEP><VDC:ORIGINX>
<SEP><VDC:ORIGINY>
<SEP><VDC:CENTRETOPX>
<SEP><VDC:CENTRETOPY>
<SEP><VDC:CENTREBOTTOMX>
<SEP><VDC:CENTREBOTTOMY>
<SEP><VDC:CENTRELEFTX>
<SEP><VDC:CENTRELEFTY>
<SEP><VDC:CENTRERIGHTX>
<SEP><VDC:CENTRERIGHTY>
<SEP><VDC:TOPLEFTX>
<SEP><VDC:TOPLEFTY>
<SEP><VDC:TOPRIGHTX>
<SEP><VDC:TOPRIGHTY>
<SEP><VDC:BOTTOMLEFTX>
<SEP><VDC:BOTTOMLEFTY>
<SEP><VDC:BOTTOMRIGHTX>
<SEP><VDC:BOTTOMRIGHTY>>

<SDR:NDCCONTOURATTR> ::= <<I:i_IX><SEP><I:1><SEP><I:STYLE>
{0 = solid, 1 = dashed, 2 = dotted, 3 = dashed-dotted,
4 = dashed-dashed-dotted}
<SEP><I:i_R><SEP><I:1><SEP><R: WIDTH>
<SEP><I:i_IX><SEP><I:1><SEP><I: CAP>
{0 = butted, 1 = rounded, 2 = square}
<SEP><I:i_IX><SEP><I:1><SEP><I: JOIN>
{0=round, 1=bevel, 2 = mitred}
<<>{for join type round, bevel}
| <<SEP><I:i_R><SEP><I:1><SEP><R: MITRED>>
{ for join type mitred}>>

```
<SDR:NDCTILINGCOMPONENTATTR> ::= <<I:i_VDC><SEP><I:2><SEP><VDC:TILINGORIGINX>
                                 <SEP><VDC:TILINGORIGINY>
                                 <SEP><I:i_E><SEP><I:1><SEP><DX|DY|DXDY|DYDX>
                               <  <<SEP><I:i_R><SEP><I:1><SEP><R:XVAULE>>{if DX}
                               |  <<SEP><I:i_R><SEP><I:1><SEP><R:YVAULE>>{if DY}
                               |  <<SEP><I:i_R><SEP><I:2><SEP><R:XVAULE>
                                  <SEP><R:YVALUE>>{if DXDY, DYDX}>>
```

Annex A
(normative)

Formal grammar

A.1 Introduction

This grammar is a formal definition of a CGM extended syntax for the NDC metafile.

A.2 Notation

The notational conventions are the same as those used in ISO/IEC 8632-1:1992/Amd.2:1995. They are reproduced here for convenience.

<symbol>	nonterminal
<SYMBOL>	terminal
<symbol>*	0 or more occurrences
<symbol>+	1 or more occurrences
<symbol>o	optional (0 or 1 occurrences)
<symbol>(n)	exactly n occurrences, n non-negative
<symbol-1> ::= <symbol-2>	symbol-1 has the syntax of symbol-2
<symbol-1> | <symbol-2>	symbol-1 or alternatively symbol-2
<symbol: meaning>	symbol with a stated meaning
{comment}	explanation of a symbol or a production

A.3 Detailed grammar

A.3.1 Metafile structure

```
<ndc metafile>          ::= <BEGIN NDC METAFILE>
                            <metafile identifier>
                            <metafile descriptor>
                            <picture>*
                            <END METAFILE>
<metafile identifier>   ::= <string fixed>
<picture>               ::= <BEGIN PICTURE>
                            <picture identifier>
                            <picture descriptor element>*
                            <BEGIN PICTURE BODY>
                            <picture content>*
                            <END PICTURE>
<picture identifier>    ::= <string fixed>
<picture content>       ::= <picture element>
<picture element>       ::= <graphical element>
                          |  <primitive attribute element>
```

A.3.2 Metafile descriptor elements

```
<metafile descriptor>   ::= <version>
                            <optional descriptor element>*
                            <element list>
<version>               ::= <METAFILE VERSION>
                            <integer>
<element list>          ::= <METAFILE ELEMENT LIST>
```

```
                                <element name>*
                          |     <element name shorthand enumerated>*
<element name shorthand
enumerated>           ::=  <VERSION NDC SET>
<optional descriptor element> ::= <VDC TYPE>
                                <vdc type enumerated>
                          |  <MAXIMUM COLOUR INDEX>
                                <colour index>
                          |  <FONT LIST>
                                <font name>+
                          |  <CHARACTER SET LIST>
                                <character set definition>+
                          |  <CHARACTER CODING ANNOUNCER>
                                <coding technique enumerated>
                          |  <scalar precision>
                          |  <MAXIMUM VDC EXTENT>
                                <point>(2)
<vdc type enumerated>     ::= <INTEGER>
                          |  <REAL>
<font name>               ::= <string fixed>
<character set enumerated> ::= <94 CHAR>
                          |  <96 CHAR>
                          |  <MULTI-BYTE 94 CHAR>
                          |  <MULTI-BYTE 96 CHAR>
                          |  <COMPLETE CODE>
<coding technique enumerated> ::= <BASIC 7-BIT>
                          |  <BASIC 8-BIT>
                          |  <EXTENDED 7-BIT>
                          |  <EXTENDED 8-BIT>
<scalar precision>        ::= <INTEGER PRECISION>
                                <integer precision value>
                          |  <REAL PRECISION>
                                <real precision value>
                          |  <INDEX PRECISION>
                                <index precision value>
                                {these elements have encoding dependent}
                                {parameters}
```

A.3.3 Picture descriptor elements

```
<picture descriptor element> ::= <VDC EXTENT>
                                <point>(2)
<point>                   ::= <vdc value>(2)
```

A.3.4 Control elements

```
<control element>         ::= <clip indicator element>
                          |  <clip rectangle element>
<clip indicator element>  ::= <CLIP INDICATOR>
                                <off-on indicator enumerated>
<off-on indicator enumerated> ::= <OFF>
                          |  <ON>
<clip rectangle element>  ::= <CLIP RECTANGLE><point>(2)
<scissor identifier element> ::= <SCISSOR IDENTIFIER>
```

```
                                <index>
<index>                 ::=     <positive integer>
<shield indicator element>  ::= <SHIELD INDICATOR>
                                    <off-on indicator enumerated>
```

A.3.5 Graphical elements

```
<graphical element>     ::=     <setofpolyline>
                        |       <setofnurb>
                        |       <setofconicsection>
                        |       <polymarker element>
                        |       <setoffillarea>
                        |       <setofellipticsector>
                        |       <setofellipticsegment>
                        |       <setofellipticdisc>
                        |       <setofclosednurb>
                        |       <text element>
                        |       <cellarray>
                        |       <design>
                        |       <gdp element>
<setofpolyline>         ::=     <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC SET OF POLYLINE>
                                    <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <polyline element>+
                                <END APPLICATION STRUCTURE>
                        |       <polyline element>
<application structure identifier>  ::= <string fixed>
<polyline element>      ::=     <POLYLINE>
                                    <point pair>
                                    <point list>
<point list>            ::=     <point>*
<point pair>            ::=     <point>(2)
<setofnurb>             ::=     <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC SET OF NURB>
                                    <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <non-uniform rational B-spline element>+
                                <END APPLICATION STRUCTURE>
                        |       <non-uniform rational B-spline element>
<non-uniform rational B-spline
element>                ::=     <NON-UNIFORM RATIONAL B-SPLINE>
                                    <spline order>
                                    <number of control points>
                                    <control points>
                                    <list of knots>
                                    <parameter start value>
                                    <parameter end value>
                                    <weights>
<spline order>          ::=     <positive integer>{m}
<number of control points>  ::= <positive integer>{n ≥ m}
```

```
<control points>            ::= <point>(n)
<list of knots>             ::= <real>(n + m)
<parameter start value>     ::= <real>
<parameter end value>       ::= <real>
<weights>                   ::= <real>(n)
<setofconicsection>         ::= <BEGIN APPLICATION STRUCTURE>
                                  <application structure identifier>
                                  <NDC SET OF CONIC SECTION>
                                  <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                  <conic section element>+
                                <END APPLICATION STRUCTURE>
                              | <conic section element>
<conic section element>     ::= <CONIC SECTION>
                                  <matrix33>
                                  <point>(2)
                                  <senseflag>
<matrix33>                  ::= <3 × 3 matrix of reals>
<senseflag>                 ::= <CLOCKWISE>
                              | <ANTICLOCKWISE>
<polymarker element>        ::= <POLYMARKER>
                                  <point>
                                  <point list>
<setoffillarea>             ::= <BEGIN APPLICATION STRUCTURE>
                                  <application structure identifier>
                                  <NDC SET OF FILL AREA>
                                  <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                  <polygon element>+
                                <END APPLICATION STRUCTURE>
                              | <polygon element>
<polygon element>           ::= <POLYGON>
                                  <point>(3)
                                  <point list>
<setofellipticsector>       ::= <BEGIN APPLICATION STRUCTURE>
                                  <application structure identifier>
                                  <NDC SET OF ELLIPTIC SECTOR>
                                  <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                  <conic section element>+
                                <END APPLICATION STRUCTURE>
<setofellipticsegment>      ::= <BEGIN APPLICATION STRUCTURE>
                                  <application structure identifier>
                                  <NDC SET OF ELLIPTIC SEGMENT>
                                  <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                  <conic section element>+
                                <END APPLICATION STRUCTURE>
<setofellipticdisc>         ::= <BEGIN APPLICATION STRUCTURE>
                                  <application structure identifier>
                                  <NDC SET OF ELLIPTIC DISC>
                                  <STATE LIST>
```

Annex A

Detailed grammar

```
                <BEGIN APPLICATION STRUCTURE BODY>
                                    <elliptic disc element>+
                                <END APPLICATION STRUCTURE>
                            |   <elliptic disc element>
<elliptic disc element>     ::= <ELLIPTIC DISC>
                                    <matrix33>
<setofclosednurb>           ::= <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC SET OF CLOSED NURB>
                                    <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <non-uniform rational B-spline element>+
                                <END APPLICATION STRUCTURE>
<text element>              ::= <TEXT>
                                    <point>
                                    <FINAL>
                                    <string>
<cellarray>                 ::= <cell array indexed colour>
                            |   <cell array direct colour>
<cell array indexed colour> ::= <COLOUR SELECTION MODE>
                                    <INDEXED>
                                <CELL ARRAY>
                                    <point>(3)
                                    <positive integer>(2)
                                    <local colour precision>
                                    <colour index>(integer1 × integer2)
                                    {This element has an encoding dependent parameter}
<positive integer>          ::= <integer> {≥ 0}
<cell array direct colour>  ::= <colour model>o
                                <COLOUR SELECTION MODE>
                                    <DIRECT>
                                <CELL ARRAY>
                                    <point>(3)
                                    <positive integer>(2)
                                    <local colour precision>
                                    <colour direct>(integer1 × integer2)
                                    {This element has an encoding dependent parameter}
<local colour precision>    ::= <colour precision value>
                            |     <colour index precision value>
                            |     <default colour precision indicator>
<colour model>              ::= <COLOUR VALUE EXTENT>
                                    <colour value mapping specifier>
                                <COLOUR PRECISION>
                                    <colour precision value>
                                    {this element has an encoding dependent parameter}
                                <COLOUR MODEL>
                                    <positive index>
<colour value mapping specifier> ::= <colour direct>(2)
                            |   (<colour scale><colour offset>)(3)
<colour scale>              ::= <real>
<colour offset>             ::= <real>
<positive index>            ::= <positive integer>
```

```
<design>                          ::= <BEGIN APPLICATION STRUCTURE>
                                      <application structure identifier>
                                      <NDC DESIGN >
                                      <STATE LIST>
                                    <APPLICATION STRUCTURE ATTRIBUTE>
                                      <NDC DESIGN PARAMETERS>
                                      <stencil origin>
                                      <stencil transformation>
                                      <tiling origin>
                                      <tiling transformation>
                                    <BEGIN APPLICATION STRUCTURE BODY>
                                      <stencil>
                                      <tiling>
                                    <END APPLICATION STRUCTURE>
<stencil origin>                  ::= <point>
<stencil transformation>          ::= <2 × 3 matrix of reals>
<tiling origin>                   ::= <point>
<tiling transformation>           ::= <2 × 3 matrix of reals>
<stencil>                         ::= <BEGIN APPLICATION STRUCTURE>
                                      <application structure identifier>
                                      <NDC STENCIL >
                                      <STATE LIST>
                                    <APPLICATION STRUCTURE ATTRIBUTE>
                                      <NDC INSIDE RULE ENUMERATED>
                                      <inside rule enumerated>
                                    <APPLICATION STRUCTURE ATTRIBUTE>
                                      <NDC STENCIL ATTRIBUTES>
                                      <stencil attributes>
                                    <BEGIN APPLICATION STRUCTURE BODY>
                                      <stencil outline>+
                                    <END APPLICATION STRUCTURE>
<inside rule enumerated>          ::= <EVENODD>
                                    | <WINDING>
<stencil attributes>              ::= <real>(9) {ordinate attributes - order as in ISO/IEC 7942-1:1994, 8.7.3}
                                      <real>(10) {coordinate attributes - order as in ISO/IEC 7942-1:1994, 8.7.3}
<stencil outline>                 ::= <path contour>
                                    | <boundary sequence>
<path contour>                    ::= <BEGIN APPLICATION STRUCTURE>
                                      <application structure identifier>
                                      <NDC SET OF PATHS >
                                      <STATE LIST>
                                    <APPLICATION STRUCTURE ATTRIBUTE>
                                      <NDC INSIDE RULE ENUMERATED>
                                      <inside rule enumerated>
                                    <APPLICATION STRUCTURE ATTRIBUTE>
                                      <NDC CONTOUR ATTRIBUTES>
                                      <contour attributes>
                                    <BEGIN APPLICATION STRUCTURE BODY>
                                      <path>+
                                    <END APPLICATION STRUCTURE>
<contour attributes>              ::= <contour style>
                                      <contour width>
```

```
                                <contour cap>
                                    <contour join>
<contour style>          ::=    <SOLID>
                            |   <DASHED>
                            |   <DOTTED>
                            |       <DASHED-DOTTED>
                            |   <DASHED-DOTTED-DOTTED>
<contour width>          ::=    <real>
<contour cap>            ::=    <BUTTED>
                            |   <ROUNDED>
                            |   <SQUARE>
<contour join>           ::=    <ROUND>
                            |   <BEVEL>
                            |   <MITRED>
                                    <real>
<path>                   ::=    <polyline element>
                            |   <non-uniform rational B-spline element>
                            |   <conic section element>
<boundary sequence>      ::=    <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC BOUNDARY SEQUENCE >
                                    <STATE LIST>
                                <APPLICATION STRUCTURE ATTRIBUTE>
                                    <NDC INSIDE RULE ENUMERATED>
                                    <inside rule enumerated>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <boundary>+
                                <END APPLICATION STRUCTURE>
<boundary>               ::=    <path>
                            |   <elliptic disc element>
<tiling>                 ::=    <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC TILING >
                                    <STATE LIST>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <tiling component>+
                                <END APPLICATION STRUCTURE>
<tiling component>       ::=    <BEGIN APPLICATION STRUCTURE>
                                    <application structure identifier>
                                    <NDC TILING COMPONENT>
                                    <STATE LIST>
                                <APPLICATION STRUCTURE ATTRIBUTE>
                                    <NDC TILING COMPONENT ATTRIBUTES>
                                    <tiling component attributes>
                                <BEGIN APPLICATION STRUCTURE BODY>
                                    <tile>+
                                <END APPLICATION STRUCTURE>
<tiling component attributes> ::=   <tiling origin>
                                <replication technique>
<replication technique>  ::=    <DX>
                                    <real>
                            |   <DY>
```

```
                        <real>
                  |   <DXDY>
                        <real>(2)
                  |   <DYDX>
                        <real>(2)
<tile>            ::= <polylinetile>
                  |   <nurbtile>
                  |   <conicsectiontile>
                  |   <fillareatile>
                  |   <ellipticsectortile>
                  |   <ellipticsegmenttile>
                  |   <ellipticdisctile>
                  |   <closednurbtile>
                  |   <design>

<polylinetile>    ::= <curvecolourspecifier>
                      <setofpolyline>
<nurbtile>        ::= <curvecolourspecifier>
                      <setofnurb>
<conicsectiontile>::= <curvecolourspecifier>
                      <setofconicsection>
<fillareatile>    ::= <interiorcolourspecifier>
                      <setoffillarea>
<ellipticsectortile>::= <interiorcolourspecifier>
                      <setofellipticsector>
<ellipticsegmenttile>::= <interiorcolourspecifier>
                      <setofellipticsegment>
<ellipticdisctile>::= <interiorcolourspecifier>
                      <setofellipticdisc>
<closednurbtile>  ::= <interiorcolourspecifier>
                      <setofclosednurb>
<gdp element>     ::= <GDP>
                        <gdp identifier>
                        <point list>
                        <data record>
<gdp identifier>  ::= <integer>
```

A.3.6 Attribute elements

```
<primitive attribute element> ::= <identification attribute>
                  |   <ndc attribute>
                  |   <area ndc attribute>
                  |   <text ndc attribute>
                  |       <curve source attribute>
                  |   <curve logical attribute>
                  |   <marker source attribute>
                  |   <marker logical attribute>
                  |   <area source attribute>
                  |   <area logical attribute>
                  |   <character source attribute>
                  |   <character logical attribute>
<identification attribute> ::= <pick identifier>
                  |   <nameset>
```

Annex A Detailed grammar

```
<pick identifier>                ::= <PICK IDENTIFIER>
                                     <name>
<nameset>                        ::= <NAMESET>
                                     <name>*
<ndc attribute>                  ::= <scissor set>
                                   | <global transformation element>
                                   | <local transformation element>
<scissorset>                     ::= <BEGIN APPLICATION STRUCTURE>
                                     <application structure identifier>
                                     <NDC SCISSOR SET >
                                     <STATE LIST>
                                     <BEGIN APPLICATION STRUCTURE BODY>
                                     <scissors>+
                                     <END APPLICATION STRUCTURE>
<scissors>                       ::= <scissor identifier element>
                                     <clip indicator element>
                                     <clip rectangle element>*
                                     <shield indicator element>
                                     <clip rectangle element>*
<global transformation element>  ::= <GLOBAL TRANSFORMATION>
                                     <matrix23>
<local transformation element>   ::= <LOCAL TRANSFORMATION>
                                     <matrix23>
<matrix23>                       ::= <2 × 3 matrix of reals>
<area ndc attribute>             ::= <pattern size>
                                   | <pattern reference point>
<pattern size>                   ::= <PATTERN SIZE>
                                     <valid size vector>(2)
<size value>                     ::= <non-negative vdc value>
                                   | <non-negative real>
<valid size vector>              ::= <valid real vector>
                                   | <valid vdc vector>
<valid real vector>              ::= <<non-zero real value>
                                     <real value>>
                                   | <<real value>
                                     <non-zero real value>
<valid vdc vector>               ::= <<non-zero vdc value>
                                     <vdc value>>
                                   | <<vdc value>
                                     <non-zero vdc value>
<non-zero vdc value>             ::= <vdc value> {greater than or less than 0}
<non-zero real value>            ::= <real value> {greater than or less than 0}
<non-negative vdc value>         ::= <vdc value> {greater than or equal to 0}
<non-negative real value>        ::= <real value> {greater than or equal to 0}
<pattern reference point>        ::= <FILL REFERENCE POINT>
                                     <point>
<text ndc attribute>             ::= <text height>
                                   | <text up vector and text skew angle>
                                   | <text path>
                                   | <text alignment>
<text height>                    ::= <CHARACTER HEIGHT>
                                     <non-negative vdc value>
```