

Informationstechnik
Graphische Datenverarbeitung
Datei für die Speicherung und Übertragung von Bildinformationen
 Teil 1: Funktionsbeschreibung
 (ISO/IEC 8632-1 : 1992) Englische Fassung EN 28632-1 : 1994

DIN
EN 28632-1

Diese Norm enthält die Internationale Norm **ISO/IEC 8632-1**

ICS 35.140

Deskriptoren: Informationstechnik, graphische Datenverarbeitung, Bildinformation, Funktionsbeschreibung, Informationsverarbeitung

Ersatz für
 DIN ISO 8632-1 : 1990-08

Information technology – Computer Graphics, Metafile for the storage and transfer of picture description information – Part 1: Functional specification;
 (ISO/IEC 8632-1 : 1992); English version EN 28632-1 : 1994

Technologies de l'information – Infographie, Métafichier de stockage et de transfert des informations de description d'images – Partie 1: Description fonctionnelle;
 (ISO/IEC 8632-1 : 1992); Version anglaise EN 28632-1 : 1994

Die Europäische Norm EN 28632-1 : 1994 hat den Status einer Deutschen Norm.

Nationales Vorwort

1 Allgemeines

Die Internationale Norm wurde von ISO/IEC JTC 1/SC 24*) unter Mitwirkung des Ausschusses "Graphische Datenverarbeitung und Bildverarbeitung" des Normenausschusses Informationsverarbeitungssysteme (NI) erarbeitet. Diese zweite Ausgabe hebt die erste (ISO 8632-1 : 1987) auf und ersetzt sie. Es sind die erarbeiteten Ergänzungen sowie Fehlerberichtigungen eingearbeitet.

Um den Gebrauch dieser Norm zu erleichtern, wurde dem englischen Originaltext eine englisch-deutsche Fachwörterliste vorangestellt.

In dieser Fachwörterliste werden den englischen Benennungen deutsche Benennungen und Erläuterungen zugeordnet. Die Zuordnung gilt nur für diese Norm und für deren Anwendungsbereich. Gleiche fremdsprachige Benennungen können auch in anderen Fachwörterlisten auftreten und dort anderen deutschen Benennungen zugeordnet sein. Deshalb hat diese Fachwörterliste keine allgemeine terminologische Verbindlichkeit.

2 Aufbau der Normenreihe DIN EN 28632

DIN EN 28632 besteht unter dem Haupttitel "Informationstechnik; Graphische Datenverarbeitung; Bilddatei zur Speicherung und zum Austausch von graphischer Information" aus vier Teilen:

- Teil 1 enthält die Funktionsbeschreibung
- Teil 2 beschreibt die Zeichenkodierung (im Sinne von byteorientierter Kodierung)
- Teil 3 beschreibt die Binärkodierung
- Teil 4 beschreibt die Klartextkodierung

Durch diesen Aufbau ist eine Trennung der Funktionalität von den drei Kodierungen erreicht, die jeweils in der Lage sind,

die volle Funktionalität des Teil 1 zu beschreiben. Übersetzung einer Kodierung in eine andere ist verlustfrei möglich, eine Rückübersetzung erzeugt aber nicht notwendigerweise den identischen Datenstrom.

3 Aufgabenbereich

Dieser Teil der Normenreihe beschreibt ein Dateiformat, das geeignet ist, Bildinformation zu speichern und wiederzugewinnen. In der Norm wird eine Menge von Elementen (bzw. Funktionen) festgelegt, die benutzt werden können, um Bilder unabhängig von Systemarchitekturen und Geräteeigenschaften zu beschreiben.

Die Bildbeschreibung schließt die Eigenschaften zur Beschreibung von statischen Bildern ein.

Die wesentlichen Punkte für die Entwicklung dieser Norm zur Beschreibung von Bilddateien sind:

- Speicherung von graphischer Information in wohldefinierter Weise durch ein graphisches System;
- Austausch von graphischer Information zwischen verschiedenen graphischen Systemen;
- Austausch von graphischer Information zwischen unterschiedlichen graphischen Geräten;
- Austausch von graphischer Information zwischen verschiedenen Datenverarbeitungsanlagen.

- *) ISO – International Organization for Standardization
 IEC – International Electrotechnical Commission
 JTC 1 – Joint Technical Committee "Information Technology"
 SC 24 – Computer Graphics and Image Processing

Fortsetzung Seite 2 bis 6 und
 2 Seiten EN sowie
 345 Seiten ISO/IEC-Original

Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.

4 Erzeugung und Interpretation von sowie Zugriff auf Bilddateien

Die Art und Weise der Erzeugung beziehungsweise Interpretation von Bilddateien werden in dieser Norm nur in Form informeller Hinweise beschrieben, wenn auch das angestrebte Ergebnis der Interpretation einer Bilddatei beschrieben wird. Mit den in der Norm beschriebenen Elementen (bzw. Funktionen) können anwendungsabhängige Daten innerhalb der Bilddatei gespeichert werden, die aber keine graphische Information beinhalten; Hinweise für diese Interpretation werden daher nicht gegeben.

Die Bilddatei ist so aufgebaut, daß sie sequentiell interpretiert werden kann. Nichtsequentieller Zugriff auf Bilder innerhalb der Bilddatei ist möglich, jedoch muß dieser durch die Softwareumgebung außerhalb der Bilddatei unterstützt werden.

5 Versionen von DIN EN 28632

DIN EN 28632 beschreibt unterschiedliche Versionen des Computer Graphics Metafile, denen eine unterschiedliche Funktionalität zugeordnet ist. Derzeit sind die Versionen 1, 2 und 3 definiert, wobei Version 1 der Internationalen Norm ISO 8632:1987 entspricht.

6 Status der Anhänge

Die Anhänge A, B, C sind Bestandteil der Norm. Die Anhänge D, E, F und G dienen ausschließlich der Information.

7 Fachwörterliste

1	anisotropic mapping	Verzerrte Abbildung
	Eine Abbildung, bei der die Maßstabsfaktoren entlang der Achsen nicht gleich sind. Winkel zwischen zwei nichtparallelen Geraden-Segmenten ändern sich bei solchen Transformationen, Kreise werden zu Ellipsen.	
2	aspect ratio	Aspektverhältnis
	Das Verhältnis von Breite zu Höhe, das zur Beschreibung der Gestalt eines Rechtecks in einem Koordinatensystem verwendet wird (z. B. für ein Gerätefenster oder ein Gerätedarstellungsfeld).	
3	aspect source flag (ASF)	Aspektanzeiger (AA)
	Ein Parameter, der anzeigt, ob ein bestimmter Aspekt eines Darstellungselements von einem Attributbündel ausgewählt wird oder ob die Attributauswahl individuell ist.	
4	aspects of primitives	Aspekte von Darstellungselementen
	Einflußgrößen, die die Variationsbreite im Aussehen eines Darstellungselements bestimmen. Einige Aspekte werden direkt durch individuelle gesetzte Attribute, andere indirekt durch eine Bündeltabelle gesteuert.	
5	attribute elements	Attributelemente
	Bilddateielemente, die Attribute von Darstellungselementen beschreiben.	
6	boundary	Grenze
	Die Grenze eines Fülldarstellungselementes (fill primitive) oder einer geschlossenen Figur ist der mathematische Ort der es beschreibenden Umringlinie. Bei der Konstruktion geschlossener Figuren definiert der mathematische Ort jedes beteiligten Liniendarstellungselementes und die Grenze jedes beteiligten Fülldarstellungselementes eine explizite Teilgrenze. Wenn ein	

	Füllobjekt über den Rand eines Klippbereichs hinausragt, werden diejenigen Stücke der originären geschlossenen Grenze, die außerhalb des wirksamen Klippbereichs liegen, verworfen, und Klipp-Grenzteile werden hinzugefügt, um die Geschlossenheit zu erreichen.	
7	brightness	Helligkeit
	Attribut für das visuelle Empfinden, ob ein Gebiet mehr oder weniger Licht auszustrahlen scheint.	
8	bundle	Bündel
	Eine Menge von Attributen, die sich auf eine der folgenden Kategorien von Darstellungselementen beziehen: Linien, Marken, Texte und Füllgebiete.	
9	bundle index	Bündelindex
	Ein Index in einer Bündeltabelle für ein bestimmtes Darstellungselement.	
10	bundle table	Bündeltabelle
	Eine Tabelle, die einem bestimmten Darstellungselement zugeordnet ist.	
11	character set	Zeichensatz
	Eine Sammlung darstellbarer Zeichen, die von den in TEXT, APPEND TEXT oder RESTRICTED TEXT vorkommenden individuellen Werten festgelegt werden. Ein Zeichensatz besteht aus dem (logischen) Alphabet und Sonderzeichen (nationale Zeichensätze, Katakana usw.) und ist unabhängig von der Schriftart, welche Aussehen und Stil des aktuell dargestellten Zeichens bestimmt.	
12	chroma	Farbintensität
	Die Buntheit eines Gebietes beurteilt im Verhältnis zur Helligkeit eines ähnlich beleuchteten weißen oder stark abstrahlenden Gebietes.	
13	chromaticity	Farbverhältnis
	Verhältnis von jedem der drei Dreier-Grundfarbwerte zu ihrer Summe. (Da die Summe der drei relativen Koordinaten gleich 1 ist, genügen zwei von ihnen, um das Farbverhältnis zu beschreiben.)	
14	CIELAB colour space	CIELAB-Farbmodell
	Ein nahezu gleichmäßiger Farbraum nach der CIE-Norm mit den orthogonalen Koordinaten L, a und b zur Angabe von Helligkeit, Bunton und Intensität. Bei CIELAB werden die Verhältnisswerte X/X_n , Y/Y_n und Z/Z_n verwendet, wobei X_n , Y_n , und Z_n die Werte von X, Y und Z für die passend gewählte Referenztafel "weiß" sind.	
15	CIELUV colour space	CIELUV-Farbmodell
	Ein nahezu gleichmäßiger Farbraum nach der CIE-Norm mit den orthogonalen Koordinaten L, u und v zur Angabe von Helligkeit, Bunton und Intensität. Die Komponenten u und v werden als Farbverhältnis angegeben, wobei die Farbanteile der Grundfarben auf die Referenzfarbe 'Weiß' bezogen sind. Abweichungen im Bunton werden über die Komponente L geregelt; wenn deren Wert kleiner wird, wird die Farbe dunkler.	

(fortgesetzt)

(fortgesetzt)

16 CIE tristimulus values	CIE-Dreier-Grundfarbwerte	Werte der drei Referenz-Farbreize in einem gegebenen CIE-Dreifarb-System; sie sind erforderlich, um die (Licht-)Farbe des betrachteten Reizes entsprechend der von der Commission Internationale de l'Eclairage (CIE) beschlossenen internationalen Norm zu beschreiben.
17 CIE uniform colour spaces	CIE-gleichmäßige Farbräume	Die zwei von der CIE (Commission Internationale de l'Eclairage) empfohlenen nahezu gleichmäßigen Farbräume CIELAB und CIELUV sind in CGM erlaubt. Diese Farbräume sind nichtlineare Transformationen des CIE-1931-XYZ-Dreifarbtraums in die angenäherten Entsprechungen von Helligkeit, Buntton und Intensität. Über kleine Distanzen nähern CIELAB und CIELUV ziemlich gut gleichmäßige Farbräume an, jeder von ihnen stellt ein nahezu gleichmäßiges Maß für die Beurteilung von Farbdifferenzen bereit.
18 clip indicator	Klippanzeiger	Schalter, der angibt, ob graphische Elemente an den Grenzen eines Klipprechtecks geklippt werden sollen oder nicht.
19 clipping mode	Klippmodus	Generischer Name für Linien-, Markierer- oder Kanten-Klippmodus. Der Klippmodus für graphische Objekte gibt an, ob das Klippen vor (LOCUS) oder nach Anwendung der Darstellungsattribute (SHAPE) durchgeführt wird oder zunächst das eine und dann das andere (LOCUS THEN SHAPE).
20 clipping rectangle	Klipprechteck	Das Klipprechteck spezifiziert einen Bereich (im virtuellen Koordinatenraum), bei dem die außerhalb liegenden Teile der graphischen Elemente entfernt werden.
21 clipping	Klippen	Das Entfernen derjenigen Teile von Darstellungselementen, die außerhalb einer angegebenen Grenze (normalerweise eines Fensters oder eines Darstellungsfeldes) liegen.
22 closed figure	Geschlossene Figur	Ein zusammengesetztes Darstellungselement, das sich wie ein Füllobjekt verhält. Es wird modelliert durch eine Folge von Linien- und/oder Fülldarstellungselementen, Kantenattributen und gewissen Steuerungselementen, eingeleitet durch BEGIN FIGURE und abgeschlossen durch END FIGURE.
23 CMYK colour space	CMYK-Farbraum	Ein Farbraum nach der Methode der Farbsubtraktion mit den Grundfarben Cyan (C), Magenta (M) und Gelb (Y; von Yellow) mit der redundanten Farbe Schwarz (K; von Black oder Key).
24 colour component	Farbkomponente	Eine der Dimensionen eines Farbraumes (siehe Farbmodell).
25 colour model	Farbmodell	Die Spezifikation eines dreidimensionalen (3D)-Farbkoordinatensystems oder ein 3D-Unterraum dieses Koordinatensystems, in dem jede darstellbare Farbnance durch einen Punkt repräsentiert ist. Einige Farbmodelle führen eine vierte, redundante, Dimension ein, um eine unabhängige Repräsentation von 'schwarz' zu ermöglichen. Für die Belange von CGM ist unter Farbmodell eines der folgenden zu verstehen: RGB, CIELAB, CIELUV, CMYK oder RGB-verwandt
26 colour selection mode	Farbauswahlmodus	Kennzeichnung, die anzeigt, ob die Farbauswahl direkt oder indiziert (Spezifizierung eines Indexes in einer Tabelle von Farbwerten) sein soll.
27 colour space	Farbraum	siehe Farbmodell
28 colour stimulus	Farbreiz	Sichtbarer Lichtstrahl, der das Auge trifft und ein Farbpfinden verursacht.
29 colour table	Farbtabelle	Eine Tabelle von (direkten) Farbwerten; sie wird benutzt, wenn der Farbauswahlmodus auf INDEXED gesetzt ist, um aufgrund eines Indexwertes den entsprechenden direkten Farbwert zu finden.
30 colour value	Farbwert	Werte eines n-tupels von Komponenten, die eine Farbe in einem gegebenen Farbmodell (Farbraum) beschreiben.
31 compound primitive	Zusammengesetztes Darstellungselement	Ein zusammengesetztes Darstellungselement ist durch eine Folge von CGM-Elementen spezifiziert, im Gegensatz zu Darstellungselementen, die nur aus einem einzelnen Element bestehen. Zusammengesetzter Text oder geschlossene Figuren sind Beispiele für zusammengesetzte Darstellungselemente im CGM.
32 Computer Graphics Interface (CGI)		Die Spezifikation für Schnittstellen zum Dialog mit graphischen Geräten.
33 Computer Graphics Metafile (CGM)		Die Spezifikation für eine Datei zur Speicherung und Übertragung von Bildinformation.
34 conjugate diameter pair	Konjugiertes Durchmesserpaar	Ein Paar von Durchmessern einer Ellipse, bei dem die Tangenten an die Ellipse in den Endpunkten des Durchmessers parallel zum anderen Durchmesser sind.
35 control elements	Steuerungselemente	Bilddateielemente zur Spezifizierung von Adreßraum, Klippgrenze, Bildbegrenzungen und Formatbeschreibungen der Bilddateielemente.
36 data interface	Datenschnittstelle	Eine Schnittstelle zwischen Softwaremodulen oder Geräten, die aus Sätzen von Operationsschlüsseln und Daten besteht (im Gegensatz zu einer Schnittstelle mit Unterprogrammaufrufen).

(fortgesetzt)

(fortgesetzt)

37	descriptor elements	Beschreibende Elemente Bilddateielemente zur Beschreibung des funktionalen Inhalts, des Formats, der Voreinstellungen, der Identifikation und spezieller Eigenschaften der Bilddatei.
38	device coordinates	Gerätekoordinaten Dem Gerät eigene Koordinaten. Andere Bezeichnungen: Geräteabhängige Koordinaten, physikalische Geräte-Koordinaten
39	device driver	Gerätetreiber Der geräteabhängige Teil einer Graphik-Implementierung zur Unterstützung eines graphischen Gerätes. Der Gerätetreiber erzeugt eine geräteabhängige Ausgabe.
40	device viewport	Geräte-Darstellungsfeld Eine rechteckige Region des Gerätekoordinatenraums, auf die die VDC-Ausdehnung (Fenster) abgebildet wird. Siehe auch 44 (effektives Darstellungsfeld).
41	direct colour	Direkte Farbuordnung Eine Farbuordnung, bei der die Farbwerte direkt spezifiziert werden, ohne daß eine Zwischenabbildung über eine Farbtabelle nötig wird. Siehe 7.29 (Farbtabelle) und 7.62 (Indizierte Farbuordnung).
42	display surface	Darstellungsfläche, Sichtfläche Der Teil des graphischen Gerätes, auf dem das Bild erscheinen kann.
43	edge	Kante Die Kante eines Füllobjektes besteht aus Teilen, jeder Teil ist ursprünglich ein Liniendarstellungselement oder die Grenze eines Fülldarstellungselementes. Die Kante schließt nicht die impliziten Teilgrenzen ein, die bei der Modellierung einer geschlossenen Figur zur Gewährleistung der Geschlossenheit hinzugefügt wurden. Jeder Teilkante sind Kantenattributwerte zugeordnet, die von Teilkante zu Teilkante verschieden sein können. Nach dem Klippen werden ganze Kantenteile oder Teile von Kantenteilen, die außerhalb des entsprechenden Klipprechteckes liegen, verworfen. Wenn die (geklippte) Kante zur Darstellung aufbereitet wird, werden die verbleibenden Stücke einer Teilkante, die nunmehr unangebunden sein können, entsprechend ihrer zugeordneten Kantenattribute aufbereitet. Siehe auch 6 (Grenze).
44	effective viewport	effektives Darstellungsfeld Das aktuelle Darstellungsfeld, das wegen der verzerrungsfreien Abbildung der VDC-Ausdehnung auf das angeforderte Geräte-Darstellungsfeld von diesem etwas abweichen kann.
45	escape elements	Verschiebungselemente Bilddateielemente, die geräte- oder implementierungsabhängige Elemente zum Aufbau eines Bildes beschreiben, die aber nicht genormt sind.
46	external elements	Externe Elemente Bilddateielemente zur Übertragung von Information, die nicht direkt zur Generierung eines Bildes gehört.
47	escapement	Verschiebung Die Verschiebung der aktuellen Position auf der Darstellungsfläche, nachdem während der Aufbereitung einer Zeichenfolge auf dem Bildschirm der Repräsentant eines Zeichens dargestellt ist.
48	escapement point	Verschiebungspunkt Ein Maßwert eines Zeichens; ein Punkt in dem Koordinatensystem des Zeichens, zu dem die aktuelle Position auf der Darstellungsfläche normalerweise verschoben wird, nachdem der Repräsentant des Zeichens dargestellt ist.
49	font	Schriftart Ein Attributwert eines Textdarstellungselementes, der bei der Darstellung das Aussehen der Zeichen bestimmt. Siehe auch Zeichensatz.
50	font family	Schriftartfamilie Mehrere Schriftarten mit gleichem Design, z. B. Courier, Courier fett, Courier kursiv.
51	font metrics	Schriftartmetrik Informationen zur Dimensionierung und Positionierung von Zeichen in einer Schriftartcharakteristik, soweit sie allen Repräsentationen von Zeichen dieser Schriftart gemeinsam sind.
52	font resource	Schriftartcharakteristik Eine Menge von Zeichen-Repräsentationen mit Beschreibung und Informationen über die Schriftartmetrik, soweit sie für die Menge als ganzer von Bedeutung sind.
53	foreground colour	Vordergrundfarbe Die durch einen zugeordneten Attributwert festgelegte Farbe, in der elementare Komponenten von Objekten oder Innenbereiche von Objekten für die Darstellung in der Zeichenfläche aufbereitet werden. Wenn ein Linientyp nicht zusammenhängend oder ein Füllmuster nicht flächenfüllend ist, wird die Hilfsfarbe (auxiliary colour) für die Zwischenräume verwendet. Bei Text werden die Zeichen in der Vordergrundfarbe dargestellt, der Rest der Zeichen-Zelle erscheint in der Hilfsfarbe.
54	glyph	Zeichen Ein wiedererkennbares abstraktes graphisches Symbol, unabhängig von seinem spezifischen Aussehen.
55	global segment	globales Segment Ein Segment, das im Metafile-Beschreibungsteil definiert ist und deshalb innerhalb von jedem Bild dieses Metafiles referenziert werden kann.
56	graphical primitive elements	graphische Darstellungselemente Darstellungselemente geben die geometrische Beschreibung für einen Teil eines Bildes.
57	Graphical Kernel System (GKS)	Graphisches Kernsystem Norm, die eine Menge von Funktionen für die Programmierung in der graphischen Datenverarbeitung spezifiziert. Siehe DIN 66252-1.
58	graphics device	Graphisches Gerät Ein Gerät, auf dem Bilder ausgegeben werden können.

(fortgesetzt)

(fortgesetzt)

59 graphic object Ein Graphikobjekt besteht aus einem oder mehreren (zusammengesetzten) graphischen Darstellungselementen, zusammen mit den zugeordneten Attributwerten.	Graphikobjekt	71 normalized device coordinates (NDC) Eine Koordinate in einem geräteunabhängigen Koordinatensystem, das auf einem Bereich – häufig auf 0 bis 1 – normiert ist. Siehe auch 94, 95, 96 und 97.	Normierte Koordinate
60 hatch style Die Signatur für den Innenbereich eines Füllgebietes mit Füllart HATCH. Eine Schraffurart besteht aus einer oder mehreren Mengen von (quasi-)äquidistanten Linien (in verschiedenen Richtungen).	Schraffurart	72 object clipping Falls erforderlich, wird das Klippen von Graphikobjekten am Klipprechteck in Abhängigkeit vom Klippmodus durchgeführt.	Objektklippen
61 hue Attribut einer visuellen Wahrnehmung, nach der die Farbe eines Gebietes Rot, Gelb, Grün oder Blau ist oder eine Kombination von zwei dieser Farben.	Bunton	73 pattern style Eine Beschreibung des Inneren eines Füllgebietes. Eine Musterart besteht aus einem Feld mit verschiedenen gefärbten oder schattierten Zellen.	Musterart
62 indexed colour Eine Farbzuoordnung, bei der der Farbindex dazu benutzt wird, die Farbwerte aus einer Tabelle zu erhalten. Siehe auch 29 (Farbtabelle) und 41 (Direkte Farbzuoordnung).	Indizierte Farbzuoordnung	74 picture Descriptor Eine Menge von Bilddateielementen zum Setzen der Art und Weise der Interpretation von Attributelementen für das gesamte Bild.	Bildbeschreibung
63 isotropic mapping Eine Abbildung, die bezüglich der Richtung invariant ist, mit gleichem Maßstab in allen orthogonalen Achsenrichtungen. Diese Transformation wird oft benutzt, um die VDC-Ausdehnung auf die physikalische Zeichenfläche abzubilden. Bei verzerrungsfreien Abbildungen bleiben die Winkel zwischen zwei nicht-parallelen Geraden erhalten, Kreise bleiben Kreise.	Verzerrungsfreie Abbildung	75 pixel Bei Rastergeräten das kleinste Element der Zeichenfläche, dem individuell eine Farbe oder ein Graustufenwert zugewiesen werden kann.	Pixel
64 local segment Ein Segment, das im Bild-Beschreibungsteil oder im Bildkörper definiert ist. Innerhalb des Metafiles kann es daher nur lokal in dem Bild referenziert werden, in dem es definiert ist. Siehe auch 55 (Globales Segment).	Lokales Segment	76 posture Das Maß, um das die Form eines oder mehrerer Zeichen geneigt aussieht.	Positur
65 message Eine Zeichenfolge, die das Anwendungsprogramm zum graphischen Arbeitsplatz sendet, zur Kommunikation mit dem Bediener.	Nachricht	77 primary colour stimuli Drei ausgewählte Lichtfarben, die verwendet werden, um die Farbe von jedem vorgegebenen Licht zu spezifizieren, und zwar durch die Beträge der drei Primärfarben, die additiv gemischt werden. (Irgendwelche drei Lichtfarben können als Primärfarben verwendet werden, wenn nicht eine von ihnen durch Mischen der beiden anderen erzeugt werden kann. Um durch additives Mischen ein Maximum an Farbabstufungen zu erreichen, werden gewöhnlich gesättigtes Rot, Grün und Blau als Primärfarben genommen).	Primär-Farbreize/Grundfarbwerte
66 metafile Eine Datei zum Speichern von graphischen Daten und von Informationen, die eine geräteunabhängige Beschreibung von einem oder mehreren Bildern beinhalten.	Bilddatei	78 primary colorants (Stoff-/Körperfarben). Eine kleine Anzahl von Farben (Farbstoffe oder Pigmente), die subtraktiv gemischt werden können, um viele Farbabstufungen zu erhalten. Die häufigsten Primärfarben sind Cyan, Magenta und Gelb.	Primärfarben
67 metafile Descriptor Ein Bilddateielement zur Beschreibung des Formats der Bilddatei (aber nicht ihrer Kodierungsmethode) und der erforderlichen Funktionalität eines Bilddateiinterpreters.	Bilddateibeschreibung	79 realized edge Begrenzungslinien eines Füllgebietes. Sie hat die Breite Null, wenn sie unsichtbar ist, und endliche Breite, wenn sie sichtbar ist.	Begrenzende Kante
68 metafile element Ein Grundelement, das zum Aufbau eines Bildes oder zur Übertragung von Information verwendet werden kann.	Bilddateielement	80 realized interior Der Teil des Füllgebietes, der sich bis zur begrenzenden Kante erstreckt.	Ausfüllung
69 metafile generator Software oder Gerät zur Erstellung einer Bilddatei.	Bilddateigenerator	81 region Jedes Füllobjekt besteht aus einer oder mehreren Regionen. Jede Region hat ihre eigene geschlossene Grenze, die durch implizite Schließung entstanden und durch Objektklippen verändert sein kann. Jeder geschlossene Polygonzug eines POLYGON SET bildet eine Region. Geschlossene Figuren können aus mehreren Regionen bestehen. Regionen können sich selbst überschneidende Grenzen haben, sie können andere Regionen überlappen oder an ihnen angrenzen.	Region
70 metafile interpreter Software oder Gerät zum Lesen und Interpretieren einer Bilddatei.	Bilddateiinterpreter		

(fortgesetzt)

(abgeschlossen)

82	RGB colour space	RGB-Farbraum	Ein Farbraum mit farbmimetrischen Koordinaten, die auf den Primärfarben/-Farbreizen Rot (R), Grün (G) und Blau (B) basieren.
83	RGB-related colour space	RGB-verwandter Farbraum	Ein dem RGB-Farbraum verwandter Farbraum, der daraus durch eine lineare (3x3) Transformation hervorgeht. (Beispiele sind YUV (PAL, SECAM), YIQ (NTSC) oder $Y_C R_C B_C$ (CCITT Videocodes and CCIR601 Studiostandard)).
84	reference colour model	Referenz-Farbmodell	Ein Basis-Farbmodell in CGM, auf das zu spezifizierende Farbmodell (RGB, CMYK, CIELUV, CIELAB oder RGB-verwandt) kalibriert werden. Das Referenz-Farbmodell ist definiert durch das CIE-1931-Standard farbmimetrische System (XYZ).
85	saturation	Sättigung	Die Sättigung beschreibt das Verhältnis der Buntheit zur Helligkeit.
86	segment	Segment	Mehrere graphische Darstellungselemente oder graphische Objekte, die als zusammengehörig gruppiert werden. Jedes Segment hat eine eindeutige Segment-Identitätskennzeichnung und kann mit einer Menge von Segmentattributen, die dem Segment als ganzem zugeordnet sind, versehen sein.
87	segment attribute	Segmentattribut	Ein Attribut, das einem Segment als ganzem zugeordnet ist, nicht einzelnen Elementen.
88	size specification mode	Größen-Spezifikationsmodus	Eine generische Bezeichnung für Linienbreiten-Spezifikationsmodus, Kantenbreiten- oder Markierergrößen-Spezifikationsmodus.
89	skewed	schief	Nicht orthogonal. Dieser Ausdruck wird in verschiedenen Zusammenhängen verwendet – bei Text, wenn die Orientierungsvektoren nicht senkrecht sind, bei Zellen einer Zellmatrix oder bei Mustern, wenn die definierenden Punkte ein nichtrechtwinkliges Parallelogramm beschreiben usw. Transformationen können ebenfalls Schiefe verursachen.
90	symbol	Symbol	Ein graphisches Objekt, das an irgendeiner Stelle im Metafile nur durch Referenzierung eingefügt wird.
91	trichromatic system	Dreifarbssystem	Ein System, in dem Farbreize in Termen von Dreier-Grundfarbwerten spezifiziert werden. Es basiert auf additivem Mischen von drei passend gewählten Referenz-Farbreizen.

92	view surface	Sichtfläche	Siehe 42.
93	virtual device	Virtuelles Gerät	Ein idealisiertes graphisches Gerät, das der Graphiksoftware oder Systemen via CGI (Computer Graphic Interface) eine Menge graphischer Fähigkeiten (durch abstrakte Funktionsbeschreibungen definiert) bietet, unabhängig von der wirklichen Realisierung auf dem einen oder anderen physikalischen Gerät.
94	virtual device coordinates	Virtuelle Koordinaten (VK)	(VDC) Absolute zweidimensionale Koordinaten zur Spezifikation einer Position im VK-Raum. Siehe 97.
95	VDC extent	VK-Ausdehnung	Ein rechteckiger Bereich innerhalb des VK-Bereichs. Siehe 96, 97.
96	VDC range	VK-Bereich	Ein rechteckiger Bereich innerhalb des VK-Raums, bestehend aus der Menge aller Koordinaten, die im angegebenen Koordinatentyp, -genauigkeit und -kodierungsformat der Bilddatei dargestellt werden können. Siehe 95, 97.
97	VDC space	VK-Raum	Ein zweidimensionaler kartesischer Koordinatenraum von beliebiger Genauigkeit und Ausdehnung. Nur ein Teilbereich des VK-Raumes, der VK-Bereich, ist in einer Bilddatei realisierbar. Siehe 94, 95 und 96.
98	weight	Schwere	Das Verhältnis von Grundstrichbreite bei einem bzw. mehreren Zeichen zur Schriftarhöhe.

8 Abkürzungen

Folgende Abkürzungen werden in sämtlichen Teilen von DIN ISO 8632 benutzt:

ASF	Aspect Source Flag	Aspektanzeiger
CDP	Conjugate Diameter Pair	konjugiertes Durchmesserpaar
CGI	Computer Graphics Interface	
CGM	Computer Graphics Metafile	
GKS	Graphical Kernel System	Graphisches Kernsystem
MD	Metafile Descriptor	Bilddateibeschreibung
NDC	Normalized Device Coordinate(s)	Normalisierte Koordinaten
PD	Picture Descriptor	Bildbeschreibung
VDC	Virtual Device Coordinate(s)	Virtuelle Koordinaten

Änderungen

Gegenüber DIN ISO 8632-1 : 1980-08 wurden folgende Änderungen vorgenommen:

- EN 28632-1: 1994 übernommen.

Frühere Ausgaben

DIN ISO 8632-1 : 1990-08

Deskriptoren: Datenverarbeitung, graphische Datenverarbeitung, Datenaustausch, Datenbearbeitung, Datenübertragung, Datendarstellung, Beschreibung, Datei

Deutsche Fassung

Informationstechnik
Graphische Datenverarbeitung
Datei für die Speicherung und Übertragung von Bildinformationen
Teil 1: Funktionsbeschreibung
(ISO/IEC 8632-1 : 1992)

Information technology – Computer Graphics,
Metafile for the storage and transfer of picture
description information – Part 1: Functional
specification
(ISO/IEC 8632-1 : 1992)

Technologies de l'information – Infographie,
Métafichier de stockage et de transfert des in-
formations de description d'images – Partie 1:
Description fonctionnelle
(ISO/IEC 8632-1 : 1992)

Diese Europäische Norm wurde von CEN am 1994-01-14 angenommen.

Die CEN-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CEN-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CEN-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CEN-Mitglieder sind die nationalen Normungsinstitute von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien und dem Vereinigten Königreich.

CEN

EUROPÄISCHES KOMITEE FÜR NORMUNG

European Committee for Standardization

Comité Européen de Normalisation

Zentralsekretariat: rue de Stassart 36, B-1050 Brüssel

Vorwort

Auf Vorschlag des Zentralsekretariats von CEN hatte das Technische Büro entschieden, die Internationale Norm:

"Informationstechnik – Graphische Datenverarbeitung, Datei für die Speicherung und Übertragung von Bildinformationen – Teil 1: Funktionsbeschreibung (ISO/IEC 8632-1 : 1992)"

zur formellen Abstimmung vorzulegen.

Das Ergebnis der formellen Abstimmung war positiv.

Zur Zeit besteht dieses Dokument nur in Englisch und Französisch.

Diese Europäische Norm muß den Status einer nationalen Norm erhalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis Juli 1994, und etwaige entgegenstehende nationale Normen müssen bis Juli 1994 zurückgezogen werden.

Entsprechend der CEN/CENELEC-Geschäftsordnung sind folgende Länder gehalten, diese Europäische Norm zu übernehmen:

Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien und das Vereinigte Königreich.

Anerkennungsnotiz

Der Text der Internationalen Norm ISO/IEC 8632-1 : 1992 wurde von CEN ohne jegliche Änderung als Europäische Norm angenommen.

**Information technology — Computer graphics —
Metafile for the storage and transfer of picture
description information —**

Part 1:
Functional specification

*Technologies de l'information — Infographie — Métafichier de stockage
et de transfert des informations de description d'images —*

Partie 1: Description fonctionnelle



Reference number
ISO/IEC 8632-1:1992(E)

CONTENTS

1	Scope	1	4.6	Graphical primitive elements	31
2	Normative references	2	4.6.1	Line elements	31
3	Definitions and abbreviations	4	4.6.2	Marker element	33
3.1	Definitions	4	4.6.3	Text elements	35
3.2	Abbreviations	9	4.6.4	Filled-area elements	36
4	Concepts	10	4.6.5	Cell elements	36
4.1	Introduction	10	4.6.6	Circular arc elements	38
4.2	Delimiter elements	10	4.6.7	Elliptical elements	41
4.3	Metafile descriptor elements	11	4.6.8	Hyperbolic Arc Element	41
4.3.1	Identification	12	4.6.9	Parabolic arc element	42
4.3.2	Functional capability	12	4.6.10	Spline curve elements	46
4.3.3	Default metafile state	16	4.6.11	Closed figures	48
4.3.4	Fonts and character sets	16	4.6.12	Symbol elements	54
4.4	Picture descriptor elements	21	4.7	Attribute elements	55
4.4.1	Scaling mode	21	4.7.1	Line attributes	58
4.4.2	Colour selection mode	21	4.7.2	Marker attributes	59
4.4.3	Specification modes	22	4.7.3	Text attributes	61
4.4.4	VDC extent	22	4.7.4	Filled-area attributes	85
4.4.5	CGM tailoring	22	4.7.5	Specification modes and transformation of aspects	87
4.4.6	Background colour	24	4.7.6	Colour attributes	88
4.4.7	Device viewport control	24	4.7.7	Pick identifier	90
4.4.8	Representations	25	4.7.8	Compound text path	90
4.4.9	Definable attributes	25	4.7.9	Symbol Attributes	91
4.5	Control elements	25	4.8	Escape elements	93
4.5.1	VDC space and range	25	4.9	External elements	93
4.5.2	Clipping	25	4.10	Segment elements	93
4.5.3	Save and restore primitive context	27	4.10.1	Introduction	94
4.5.4	Compound clipping and shielding	27	4.10.2	Local and global segments	94
4.5.5	Generalized text path	30	4.10.3	Delimiting and naming segments	95
4.5.6	Mitre limit	30	4.10.4	Segment attributes	95
4.5.7	Transparent cell colour	30	4.10.5	Copy segment and inheritance	96
			4.11	Metafile states	101
			4.12	Registration	112
			5	Abstract specification of elements	115
			5.1	Introduction	115
			5.2	Delimiter elements	119
			5.2.1	BEGIN METAFILE	119
			5.2.2	END METAFILE	119
			5.2.3	BEGIN PICTURE	119
			5.2.4	BEGIN PICTURE BODY	120
			5.2.5	END PICTURE	121
			5.2.6	BEGIN SEGMENT	121
			5.2.7	END SEGMENT	121
			5.2.8	BEGIN FIGURE	122
			5.2.9	END FIGURE	122
			5.2.10	BEGIN PROTECTION REGION	122
			5.2.11	END PROTECTION REGION	123
			5.2.12	BEGIN COMPOUND LINE	123
			5.2.13	END COMPOUND LINE	123
			5.2.14	BEGIN COMPOUND TEXT PATH	124
			5.2.15	END COMPOUND TEXT PATH	124
			5.2.16	BEGIN TILE ARRAY	124
			5.2.17	END TILE ARRAY	125

5.3	Metafile descriptor elements	127
5.3.1	METAFILE VERSION	127
5.3.2	METAFILE DESCRIPTION	127
5.3.3	VDC TYPE	127
5.3.4	INTEGER PRECISION	128
5.3.5	REAL PRECISION	128
5.3.6	INDEX PRECISION	128
5.3.7	COLOUR PRECISION	129
5.3.8	COLOUR INDEX PRECISION	129
5.3.9	MAXIMUM COLOUR INDEX	129
5.3.10	COLOUR VALUE EXTENT	130
5.3.11	METAFILE ELEMENT LIST	131
5.3.12	METAFILE DEFAULTS REPLACEMENT	132
5.3.13	FONT LIST	132
5.3.14	CHARACTER SET LIST	132
5.3.15	CHARACTER CODING ANNOUNCER	133
5.3.16	NAME PRECISION	134
5.3.17	MAXIMUM VDC EXTENT	134
5.3.18	SEGMENT PRIORITY EXTENT	134
5.3.19	COLOUR MODEL	135
5.3.20	COLOUR CALIBRATION	135
5.3.21	FONT PROPERTIES	137
5.3.22	GLYPH MAPPING	141
5.3.23	SYMBOL LIBRARY LIST	142
5.4	Picture descriptor elements	143
5.4.1	SCALING MODE	143
5.4.2	COLOUR SELECTION MODE	143
5.4.3	LINE WIDTH SPECIFICATION MODE	144
5.4.4	MARKER SIZE SPECIFICATION MODE	144
5.4.5	EDGE WIDTH SPECIFICATION MODE	145
5.4.6	VDC EXTENT	145
5.4.7	BACKGROUND COLOUR	146
5.4.8	DEVICE VIEWPORT	146
5.4.9	DEVICE VIEWPORT SPECIFICATION MODE	147
5.4.10	DEVICE VIEWPORT MAPPING	147
5.4.11	LINE REPRESENTATION	148
5.4.12	MARKER REPRESENTATION	148
5.4.13	TEXT REPRESENTATION	149
5.4.14	FILL REPRESENTATION	150
5.4.15	EDGE REPRESENTATION	150
5.4.16	INTERIOR STYLE SPECIFICATION MODE	151
5.4.17	LINE AND EDGE TYPE DEFINITION	151
5.4.18	HATCH STYLE DEFINITION	152
5.4.19	GEOMETRIC PATTERN DEFINITION	153
5.5	Control elements	155
5.5.1	VDC INTEGER PRECISION	155
5.5.2	VDC REAL PRECISION	155
5.5.3	AUXILIARY COLOUR	155
5.5.4	TRANSPARENCY	156
5.5.5	CLIP RECTANGLE	157
5.5.6	CLIP INDICATOR	157
5.5.7	LINE CLIPPING MODE	157
5.5.8	MARKER CLIPPING MODE	158
5.5.9	EDGE CLIPPING MODE	158
5.5.10	NEW REGION	158
5.5.11	SAVE PRIMITIVE CONTEXT	159
5.5.12	RESTORE PRIMITIVE CONTEXT	160
5.5.13	PROTECTION REGION INDICATOR	161
5.5.14	GENERALIZED TEXT PATH MODE	161
5.5.15	MITRE LIMIT	161
5.5.16	TRANSPARENT CELL COLOUR	161
5.6	Graphical primitive elements	163
5.6.1	POLYLINE	163
5.6.2	DISJOINT POLYLINE	163
5.6.3	POLYMARKER	163
5.6.4	TEXT	164
5.6.5	RESTRICTED TEXT	165
5.6.6	APPEND TEXT	166
5.6.7	POLYGON	167
5.6.8	POLYGON SET	168
5.6.9	CELL ARRAY	171
5.6.10	GENERALIZED DRAWING PRIMITIVE (GDP)	173
5.6.11	RECTANGLE	173
5.6.12	CIRCLE	174
5.6.13	CIRCULAR ARC 3 POINT	174
5.6.14	CIRCULAR ARC 3 POINT CLOSE	175
5.6.15	CIRCULAR ARC CENTRE	177
5.6.16	CIRCULAR ARC CENTRE CLOSE	177
5.6.17	ELLIPSE	178
5.6.18	ELLIPTICAL ARC	179
5.6.19	ELLIPTICAL ARC CLOSE	180
5.6.20	CIRCULAR ARC CENTRE REVERSED	181
5.6.21	CONNECTING EDGE	181
5.6.22	HYPERBOLIC ARC	182
5.6.23	PARABOLIC ARC	182
5.6.24	NON-UNIFORM B-SPLINE	183
5.6.25	NON-UNIFORM RATIONAL B-SPLINE	183
5.6.26	POLYBEZIER	184
5.6.27	POLYSYMBOL	185
5.6.28	BITONAL TILE	185
5.6.29	TILE	187
5.7	Attribute elements	190
5.7.1	LINE BUNDLE INDEX	190
5.7.2	LINE TYPE	190
5.7.3	LINE WIDTH	191
5.7.4	LINE COLOUR	192
5.7.5	MARKER BUNDLE INDEX	192
5.7.6	MARKER TYPE	192
5.7.7	MARKER SIZE	193
5.7.8	MARKER COLOUR	194
5.7.9	TEXT BUNDLE INDEX	194
5.7.10	TEXT FONT INDEX	195
5.7.11	TEXT PRECISION	195
5.7.12	CHARACTER EXPANSION FACTOR	196
5.7.13	CHARACTER SPACING	196
5.7.14	TEXT COLOUR	197

5.7.15	CHARACTER HEIGHT	197
5.7.16	CHARACTER ORIENTATION	198
5.7.17	TEXT PATH	199
5.7.18	TEXT ALIGNMENT	199
5.7.19	CHARACTER SET INDEX	200
5.7.20	ALTERNATE CHARACTER SET INDEX	200
5.7.21	FILL BUNDLE INDEX	201
5.7.22	INTERIOR STYLE	201
5.7.23	FILL COLOUR	202
5.7.24	HATCH INDEX	202
5.7.25	PATTERN INDEX	203
5.7.26	EDGE BUNDLE INDEX	204
5.7.27	EDGE TYPE	204
5.7.28	EDGE WIDTH	205
5.7.29	EDGE COLOUR	205
5.7.30	EDGE VISIBILITY	206
5.7.31	FILL REFERENCE POINT	206
5.7.32	PATTERN TABLE	207
5.7.33	PATTERN SIZE	207
5.7.34	COLOUR TABLE	208
5.7.35	ASPECT SOURCE FLAGS	209
5.7.36	PICK IDENTIFIER	210
5.7.37	LINE CAP	210
5.7.38	LINE JOIN	211
5.7.39	LINE TYPE CONTINUATION	211
5.7.40	LINE TYPE INITIAL OFFSET	212
5.7.41	TEXT SCORE TYPE	212
5.7.42	RESTRICTED TEXT TYPE	213
5.7.43	INTERPOLATED INTERIOR	214
5.7.44	EDGE CAP	215
5.7.45	EDGE JOIN	216
5.7.46	EDGE TYPE CONTINUATION	216
5.7.47	EDGE TYPE INITIAL OFFSET	217
5.7.48	SYMBOL LIBRARY INDEX	217
5.7.49	SYMBOL COLOUR	218
5.7.50	SYMBOL SIZE	218
5.7.51	SYMBOL ORIENTATION	219
5.8	Escape elements	220
5.8.1	ESCAPE	220
5.9	External elements	221
5.9.1	MESSAGE	221
5.9.2	APPLICATION DATA	221
5.10	Segment elements	222
5.10.1	COPY SEGMENT	222
5.10.2	INHERITANCE FILTER	222
5.10.3	CLIP INHERITANCE	224
5.10.4	SEGMENT TRANSFORMATION	224
5.10.5	SEGMENT HIGHLIGHTING	224
5.10.6	SEGMENT DISPLAY PRIORITY	225
5.10.7	SEGMENT PICK PRIORITY	225
6	Metafile defaults	227

7	Conformance	232
7.1	Forms of conformance	232
7.2	Functional conformance of metafiles	232
7.3	Full conformance of metafiles	232
7.4	Conformance of other encodings	232
A	Formal grammar of the functional specification of version 1 metafiles	233
A.1	Introduction	233
A.2	Notation used	233
A.3	Detailed grammar	233
A.3.1	Metafile structure	233
A.3.2	Metafile descriptor elements	234
A.3.3	Picture descriptor elements	236
A.3.4	Control elements	236
A.3.5	Graphical elements	237
A.3.6	Attribute elements	240
A.3.7	Escape elements	243
A.3.8	External elements	243
A.4	Terminal symbols	244
B	Formal Grammar of the functional specification of version 2 metafiles	248
B.1	Introduction	248
B.2	Notation used	248
B.3	Detailed grammar	248
B.3.1	Metafile structure	248
B.3.2	Metafile descriptor elements	249
B.3.3	Picture descriptor elements	252
B.3.4	Control elements	254
B.3.5	Graphical elements	255
B.3.6	Attribute elements	257
B.3.7	Closed figure element	261
B.3.8	Escape elements	261
B.3.9	External elements	262
B.3.10	Segment elements	262
B.4	Terminal symbols	264
C	Formal grammar of the functional specification of version 3 metafiles	272
C.1	Introduction	272
C.2	Definitions	272
C.2.1	Notation Used	272
C.2.2	Structured Data Records:	272
C.3	Detailed Grammar	273
C.3.1	Metafile structure	273
C.3.2	Metafile descriptor elements	278
C.3.3	Picture descriptor elements	281
C.3.4	Control elements	285
C.3.5	Graphical elements	286
C.3.6	Attribute elements	289
C.3.7	Escape elements	294
C.3.8	External elements	294
C.3.9	Segment elements	294
C.4	Terminal symbols	297
D	Guidelines for metafile generators and interpreters	307
D.1	Introduction	307

D.2	Errors and degeneracies	307
D.2.1	Syntax errors	308
D.2.2	Geometrically degenerate primitives	308
D.2.3	Mathematical singularities and ambiguities	309
D.3	General guidelines	309
D.3.1	Indexes	309
D.3.2	Colour model	312
D.3.3	Order of metafile descriptor elements	312
D.3.4	Unsatisfied references	312
D.4	Guidelines for element classes	312
D.4.1	Delimiter elements	312
D.4.2	Metafile descriptor elements	312
D.4.3	Picture descriptor elements	313
D.4.4	Control elements	313
D.4.5	Graphical primitive elements	316
D.4.6	Attribute elements	318
D.4.7	Escape elements	318
D.4.8	External elements	318
D.4.9	Segment elements	320
E	Guidelines for private encodings	321
F	Reference models	325
G	Conversions between the CIEXYZ reference colour space and metafile colour spaces	325
G.1	Introduction	325
G.2	Definitions	325
G.3	CIELUV	325
G.3.1	Conversion from the CIEXYZ reference colour space to CIELUV	326
G.3.2	Conversion from CIELUV to the CIEXYZ reference colour space	327
G.4	CIELAB	327
G.4.1	Conversion from the CIEXYZ reference colour space to CIELAB	328
G.4.2	Conversion from CIELAB to the CIEXYZ reference colour space	329
G.5	RGB	329
G.5.1	Conversion from the CIEXYZ reference colour space to RGB	330
G.5.2	Conversion from RGB to the CIEXYZ reference colour space	331
G.6	RGB-related	331
G.7	CMYK	331
G.7.1	Conversion from CMYK to the CIEXYZ reference colour space	331
G.7.2	CMYK Calibration data	331
G.8	Bibliography	332

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 JTC 1. 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 8632-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology.

This second edition cancels and replaces the first edition (ISO 8632-1:1987), which has been technically revised.

ISO/IEC 8632 consists of the following parts, under the general title *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information* :

Part 1: Functional specification

Part 2: Character encoding

Part 3: Binary encoding

Part 4: Clear text encoding

Annexes A, B, and C form an integral part of this part of ISO/IEC 8632. Annexes D, E, F and G are for information only.

Design requirements**Introduction**

- e) ISO/IEC 8632 should address the needs of different applications that have conflicting requirements for size of metafile, speed of generation and interpretation, readability, editability and ease of transfer through different transport mechanisms.

0.4 Design criteria

The requirements of 0.3 were used to formulate the following criteria which were used to decide between different design possibilities.

- a) Completeness: In any area of ISO/IEC 8632, the functionality specified by ISO/IEC 8632 should be complete in itself.
- b) Conciseness: Redundant elements or parameters should be avoided.
- c) Consistency: Contradictory elements should be avoided.
- d) Extensibility: The ability to add new elements and generality to ISO/IEC 8632 should not be precluded.
- e) Fidelity: The minimal results and characteristics of elements should be well defined.
- f) Implementability: An element should be able to be supported efficiently on most host systems and/or graphics hardware.
- g) Orthogonality: The elements of the metafile should be independent of each other, or any dependencies should be structured and well defined.
- h) Predictability: ISO/IEC 8632 should be such that the recommended or proper use of standard elements guarantees the results of using a particular element.
- i) Standard practice: Only those elements that reflect existing practice, that are necessary to support existing practice, or that are necessary to support proposed standards should be standardized.
- j) Usefulness: Functions should be powerful enough to perform useful tasks.
- k) Well-structured: The assumptions that elements make about each other should be minimized. An element should have a well-defined interface and a simply stated unconditional purpose. Multipurpose elements and side effects should be avoided.

0.5 Access to a metafile

The metafile has been designed so that, although its main usage is anticipated as being with completely sequential access, non-sequential access is also possible. Once the basic environment of the metafile has been established, individual pictures may be accessible if the medium, the encoding and the implementation support this form of access.

0.6 Generation and interpretation of metafiles

The specific mechanisms of metafile generation and interpretation are not described by ISO/IEC 8632, although it does describe the intended result of such interpretation. The basic set of metafile elements includes a capability for the addition of application-dependent data, which do not have graphical meaning and for which no intended interpretation results are described.

0.7 Distinction between formal specification and encodings

The functionality provided by the metafile is separated from the specification of any particular encoding format. ISO/IEC 8632 provides for both standard and private encodings of the elements described in this

Introduction**0.1 Purpose**

The Computer Graphics Metafile provides a file format 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 picture description includes the capability for describing static pictures. Static pictures are those where elements which may lead to dynamic effects (for example those leading to regeneration) are prohibited within the picture body.

0.2 Reasons for this International Standard

The main reasons for producing a standard computer graphics metafile are

- a) to allow picture information to be stored in an organized way on a graphical software system;
- b) to facilitate transfer of picture information between different graphical software systems;
- c) to enable picture information to be transferred between graphical devices;
- d) to enable picture information to be transferred between different computer graphics installations.

0.3 Design requirements

To reach these objectives, a number of design principles were adopted:

- a) The metafile should provide a suitable set of elements for the transfer of a wide range of pictorial information.
- b) The metafile should address the more usual and essential features found on graphical devices directly and should provide access to less common facilities via an escape mechanism.
- c) The design of the metafile should not preclude extension of ISO/IEC 8632 at a later stage to cover facilities beyond those included in this version of the Standard. It should also not preclude further extensions to support future standards.
- d) The metafile should be usable from GKS (Graphical Kernel System — ISO 7942) with both metafile input and metafile output functions. It should include the capability to support ISO 7942 (GKS) static picture capture.

Versions	Introduction
0.9 Versions ISO/IEC 8632 defines several versions of the Computer Graphics Metafile. A version is defined by a formal grammar and additional specifications contained in clauses 4, 5, and 6. Clause 7 contains conformance criteria defined by version. These versions are currently defined: Version 1 (one); Version 2 (two); and Version 3 (three).	
NOTES 1 A valid Version 2 metafile is also a valid Version 3 metafile. A valid Version 1 metafile is also a valid Version 2 metafile. 2 Version 1 metafiles are as defined by the original CGM standard, which was designated ISO 8632:1987. Version 2 metafiles are as defined by the first amendment to the CGM standard, which was designated ISO 8632:1987/Amd.1:1990. Version 3 metafiles are as defined by an amendment to Version 2 of the CGM standard. This amendment was originally designated ISO/IEC 8632:1987/Amd.3:1991, but was never published as an amendment. Instead all documents were consolidated to produce this revision of ISO 8632; Versions 1, 2, and 3 are all defined by this revision. 3 Many of the figures in this International Standard were produced using ISO/IEC 8632 standard Version 1 metafiles, as defined herein.	

Introduction	Distinction between formal specification and encodings
part of ISO/IEC 8632. Guidelines for private encodings are specified in annex E; these guidelines do not form part of ISO/IEC 8632. Three standard encodings are specified in parts 2, 3 and 4 of ISO/IEC 8632. Each of the standardized encodings is capable of representing the full functionality described in this part of ISO/IEC 8632. Translation between the standardized encodings is possible without loss of picture information, although subsequent translation back into the original encoding may not result in precisely the same data stream, due to different quantizations of precisions in the different encodings. The character encoding specified in ISO/IEC 8632-2 is intended to provide an encoding of minimum size. It conforms to the rules for code extension specified in ISO 2022 in the category of complete code system. It is particularly suitable for transfer through networks that cannot support binary transfers. The binary encoding specified in ISO/IEC 8632-3 provides an encoding that requires least effort to generate and interpret on many systems. The clear text encoding specified in ISO/IEC 8632-4 provides an encoding that can be created, viewed and edited with standard text editors. It is therefore also suitable for transfer through networks that support only transfer of text files.	0.8 Relationship to other International Standards ISO/IEC 8632 draws extensively for its model of a graphics system on GKS (Graphical Kernel System — ISO 7942). In addition, ISO/IEC 8632 specifies a metafile that may be used as a static picture-capture metafile by GKS. This part of ISO/IEC 8632 uses font concepts and the font architecture defined in ISO/IEC 9541-1 for defining CGM references to fonts and font resources. The font properties of ISO/IEC 9541-1 are adopted, where appropriate, to define CGM mechanisms to provide information useful for font substitution between parties interchanging Metafiles. This part of ISO/IEC 8632 includes the minimum font description subset defined in ISO/IEC 9541-2. Clause 3 contains a number of glossary definitions that are taken from, and are identical to, those found in ISO/IEC 9541-1. This part of ISO/IEC 8632 also defines access to extended families of glyph based on the principles and procedures of ISO/IEC 10036. This part of ISO/IEC 8632 uses a colorimetrically precise reference colour space to allow for interchange of precise colour specifications. It uses concepts defined in ISO/IEC 8613/Amd.2 which are based on CIE publications. ISO/IEC 8613/Amd.2 provides tutorial material on relevant definitions and colour concepts, which is useful for understanding the material in this Standard but is not incorporated into this Standard in that amount of detail. The character encoding specified in ISO/IEC 8632-2 conforms to the code extension techniques of ISO 2022. The binary encoding specified in ISO/IEC 8632-3 employs the mechanism for representing floating point numbers specified in ANSI/IEEE 754-1986. For certain elements, the CGM defines value ranges of parameters as being reserved for registration. The meanings of these values will be defined using the established procedures (see 4.12) of the ISO International Registration Authority for Graphical Items. These procedures do not apply to values and value ranges defined as being reserved for private use; these values and ranges are not standardized. There is a very close relationship between many of the elements in ISO 8632 and a subset of the functions in the CGI (Computer Graphics Interface — ISO/IEC 9636).

Information technology – Computer graphics – Metafile for the storage and transfer of picture description information –

Part 1 : Functional specification

1 Scope

ISO/IEC 8632 provides a file format suitable for the storage and retrieval of picture description information. The file format consists of an ordered 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 picture description includes the capability for describing static images.

The elements specified provide for the representation of a wide range of pictures on a wide range of graphical devices. The elements are split into groups that delimit major structures (metafiles and pictures), that specify the representations used within the metafile, that control the display of the picture, that perform basic drawing actions, that control the attributes of the basic drawing actions and that provide access to non-standard device capabilities.

The Metafile is defined in such a way that, in addition to sequential access to the whole metafile, random access to individual pictures is well-defined; whether this is available in any system that uses ISO/IEC 8632 depends on the medium, the encoding and the implementation.

In addition to a functional specification, three standard encodings of the metafile syntax are specified. These encodings address the needs of applications that require minimum metafile size, minimum effort to generate and interpret, and maximum flexibility for a human reader or editor of the metafile.

This part of ISO/IEC 8632 describes the format using an abstract syntax. The remaining three parts of ISO 8632 specify three standardized encodings that conform to this syntax: ISO/IEC 8632-2 specifies a character encoding that conforms to the rules for code extension specified in ISO 2022 in the category of complete coding system; ISO/IEC 8632-3 specifies a binary encoding; ISO/IEC 8632-4 specifies a clear text encoding.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 8632. 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 8632 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

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

ISO 2022:1986, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*

ISO 2375:1985, *Data processing – Procedure for registration of escape sequences.*

ISO 7942:1985, *Information processing systems – Computer graphics – Graphical Kernel System (GKS) functional description.*

ISO/IEC 8613:1989/Amd.1:–1), *Information processing systems – Text and office systems – Office document architecture (ODA) and interchange format – Amendment 1: Tiled raster graphics.*

ISO/IEC 8613:1989/Amd.2:–1), *Information processing systems – Text and office systems – Office document architecture (ODA) and interchange format – Amendment 2: Colour.*

ISO/IEC 9541-1:1991, *Information technology – Font information interchange – Part 1: Architecture.*

ISO/IEC 9541-2:1991, *Information technology – Font information interchange – Part 2: Interchange format.*

ISO/IEC 9636-1:1991, *Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices (CGI) – Functional specification – Part 1: Overview, profiles, and conformance.*

ISO/IEC 9636-2:1991, *Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices (CGI) – Functional specification – Part 2: Control.*

ISO/IEC 9636-3:1991, *Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices (CGI) – Functional specification – Part 3: Output.*

ISO/IEC 9636-4:1991, *Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices (CGI) – Functional specification – Part 4: Segments.*

ISO/IEC TR 9973:1988, *Information processing – Computer graphics – Procedures for registration of graphical items.*

ISO/IEC 10036:–1), *Information technology – Font information interchange – Procedure for registration of glyph and glyph collection identifiers.*

1) To be published.

Normative references

- CCITT Recommendation T.4 (1988), *Standardization of group 3 facsimile apparatus for document transmission*.
- CCITT Recommendation T.6 (1988), *Standardization of group 4 facsimile apparatus for document transmission*.
- CIE Publication 15.2, *Colorimetry*, 1986 (2nd edition).
- CIE Publication S002, *Colorimetric Observers*, 1986 (1st edition).

3 Definitions and abbreviations

3.1 Definitions

For the purposes of ISO/IEC 8632 the following definitions apply.

NOTE — As far as possible, commonly accepted graphics terminology is used.

- 3.1.1 anisotropic mapping:** A mapping in which the scale factors applied along each axis are not equal. This is often used in reference to the mapping from VDC to distance units on the physical display surface. With anisotropic mapping, the angle between any pair of non-parallel line segments can change; circles cease to be circles and become ellipses after such a transformation. See "isotropic mapping".
- 3.1.2 aspect ratio:** The ratio of the width to the height of a rectangular area, such as a window or viewport. For example, an aspect ratio of 2.0 indicates an area twice as wide as it is high.
- 3.1.3 aspect source flag (ASF):** Indicator as to whether a particular attribute selection is to be individual or bundled.
- 3.1.4 aspects of primitives:** Ways in which the appearance of a primitive can vary. Some aspects are controlled directly by primitive attributes; some are controlled indirectly through a bundle table.
- 3.1.5 attribute elements:** Metafile elements that describe the appearance of graphical elements.
- 3.1.6 boundary:** The mathematical locus that defines, in abstract VDC space, the limits of a region to be filled (for fill primitives and closed figures). The visual appearance of interior style 'hollow' consists of a depiction of the boundary obtained after clipping has been taken into account.
- 3.1.7 brightness:** Attribute of a visual sensation according to which an area appears to emit more or less light.
- 3.1.8 bundle:** Set of attributes associated with one of the following graphical element types: line, marker, text, and filled area.
- 3.1.9 bundle index:** An index into a bundle table for a particular output primitive.
- 3.1.10 bundle table:** An indexed table containing a set of attributes for each index.
- 3.1.11 character set:** The set of displayable symbols mapped to individual characters in a TEXT, APPEND TEXT, or RESTRICTED TEXT string. This corresponds to the "G-set" defined in ISO 2022. A character set is independent of the font or typeface; examples of character sets are: ASCII (X3.4), German and Kana-kana.
- 3.1.12 chroma:** Colourfulness (chromaticness) of an area judged in proportion to the brightness of a similarly illuminated area that appears white or highly transmitting.
- 3.1.13 chromaticity:** Ratio of each of a set of three tristimulus values to their sum. (As the sum of the three chromaticity coordinates equals 1, two of them are sufficient to define a chromaticity.)
- 3.1.14 CIELAB colour space:** A CIE recommended, approximately uniform, colour space with rectangular coordinates L^* , a^* , and b^* . L^* is the approximate correlate of lightness, a^* and b^* are used to calculate approximate correlates of hue and chroma. CIELAB uses the tristimulus ratios X/Y_n , Y/Y_n , Z/Z_n instead of chromaticity coordinates as in CIELUV. X_n , Y_n , Z_n are the values of X , Y , Z for the appropriate chosen reference white. The reduced perceptual significance of a given difference in chromaticity as the colour becomes darker is incorporated by using the tristimulus ratios.
- 3.1.15 CIELUV colour space:** A CIE recommended, approximately uniform, colour space with rectangular coordinates L^* , u^* , and v^* . L^* is the approximate correlate of lightness, u^* and v^* are used to calculate approximate correlates of hue and chroma. The colour stimulus is described by Y , u^* , v^* and the reference white by Y_n , u_n^* , v_n^* . A given difference in chromaticity is reduced in magnitude by the factor L^* as the

Definitions and abbreviations

Definitions

colour becomes darker.

3.1.16 CIE tristimulus values: Amounts of the three reference colour stimuli, in a given trichromatic system, required to match the colour of the stimulus considered.

3.1.17 CIE uniform colour spaces: Two CIE recommended, approximately uniform colour spaces, CIELAB and CIELUV, are allowed in the CGM. These colour spaces are non-linear transformations of the CIE 1931 XYZ tristimulus space, into the approximate perceptual correlates of lightness, hue, and chroma. CIELAB and CIELUV closely approximate uniform colour spaces over small distances, and they each provide an approximately uniform measure of perceived colour differences. Both colour spaces allow for the perceptual effect that a given difference in chromaticity represents a smaller and smaller colour difference as the lightness is reduced.

3.1.18 clip indicator: Indicator as to whether metafile graphical elements are to be clipped at the limits of CLIP RECTANGLE.

3.1.19 clipping mode: A generic term referring to one of Line Clipping, Marker Clipping or Edge Clipping Modes. An object clipping may be either 'focus', 'shape' or 'locus then shape'.

3.1.20 clip rectangle: A rectangle defined in VDC space which is used as a clipping boundary when the metafile graphical elements are to be clipped.

3.1.21 clipping: Removing parts of display elements that lie outside a given boundary.

3.1.22 closed figure: A compound primitive that behaves as a fill primitive of more general shape. It is formed by delimiting a sequence of line or fill primitives, edge attributes, and certain control elements, with the elements BEGIN FIGURE and END FIGURE.

3.1.23 CMYK colour space: A colour space based on the subtractive colour mixture of Cyan (C), Magenta (M) and Yellow (Y) primaries with the inclusion of black (K).

3.1.24 colour component: One of the dimensions of a colour space.

3.1.25 colour model: A specification of a 3D colour coordinate system and a 3D subspace in the coordinate system within which each displayable colour is represented by a point. Some colour models include a fourth, redundant, dimension to allow the independent representation of black. For the purpose of ISO/IEC 8632 colour model refers to one of RGB, CIELAB, CIELUV, CMYK, or RGB-related.

3.1.26 colour selection mode: Indicator as to whether colour selection is to be direct (by specifying a colour value) or indexed (by specifying an index into a table of colour values). See COLOUR VALUE.

3.1.27 colour space: See COLOUR MODEL.

3.1.28 colour stimulus: Visible radiation entering the eye and producing a sensation of colour.

3.1.29 colour table: A table for use in mapping from a colour index to the corresponding colour. See DIRECT COLOUR, INDEXED COLOUR.

3.1.30 colour value: Value of the n-tuple of components describing a colour in a given colour model.

3.1.31 compound primitive: A compound primitive is specified by a sequence of CGM elements, as opposed to primitives represented by a single element. Compound text and closed figures are examples of compound primitives in the CGM.

3.1.32 Computer Graphics Interface (CGI): The specification for interface techniques for dialogues with graphical devices.

3.1.33 Computer Graphics Metafile (CGM): The specification for a mechanism for storing and transferring picture description information.

Definitions

Definitions and abbreviations

3.1.34 conjugate diameter pair (CDP): A pair D,d of diameters of an ellipse such that a tangent to the ellipse at each endpoint of one diameter is parallel to the other diameter.

3.1.35 control elements: Metafile elements that specify address space, clipping boundaries, picture delimiters, and format descriptions of the Metafile elements.

3.1.36 data interface: An interface between software modules or devices comprising one or more packets containing opcodes and data (as contrasted with a subroutine or function call interface).

3.1.37 descriptor elements: Metafile elements that describe the functional content, format, default conditions, identification, and characteristics of the Metafile.

3.1.38 device coordinates: The coordinates native to a device; device-dependent coordinates; physical device coordinates.

3.1.39 device driver: The device-dependent part of a graphics implementation which supports a physical device. The device driver generates device-dependent output.

3.1.40 device viewport: A rectangular subset of the physical display surface into which VDC EXTENT is mapped. See "effective viewport".

3.1.41 direct colour: A colour selection scheme in which the colour values are specified directly, without requiring an intermediate mapping via a colour table. See COLOUR TABLE, INDEXED COLOUR.

3.1.42 display surface: That part of a graphics device upon which a visible image appears (for example, the screen of a display, the paper in a plotter).

3.1.43 edge: The rendering of the perimeter of a filled region, controlled by edge attributes. Edges are clipped after being applied to the boundary, as distinct from the rendition of the boundary obtained from interior style 'hollow'. See "boundary".

3.1.44 effective viewport: The actual viewport resulting from forced isotropic mapping from the VDC extent to the viewport.

3.1.45 escape elements: Metafile elements that describe device- or system-dependent elements used to construct a picture, but that are not otherwise standardized.

3.1.46 external elements: Metafile elements that communicate information not directly related to the generation of a graphical image.

3.1.47 escapement: During the rendering of text strings onto a display, the movement of the current position on the display surface after a glyph representation is imaged.

3.1.48 escapement point: A glyph metric; a point in the glyph coordinate system, to which the current position on the display surface is usually translated, after the glyph representation is imaged.

3.1.49 font: A collection of glyph images having the same basic design, e.g., *Courier Bold Oblique*.

3.1.50 font family: A collection of fonts of common design, e.g., *Courier, Courier Bold, Courier Bold Oblique*.

3.1.51 font metrics: The set of dimensions and positioning information in a font resource common to all glyph representations contained in that font resource.

3.1.52 font resource: A collection of glyph representations together with descriptive and font metric information which are relevant to the collection of glyph representations as a whole.

3.1.53 foreground colour: The colour used in the rendering process in which primitives are rendered on the display surface, as opposed to the BACKGROUND COLOUR or AUXILIARY COLOUR. The foreground colour is set separately for each class of primitive.

ISO/IEC 8632-1:1992 (E)	Definitions and abbreviations
	3.1.75 pixel: The smallest element of a display surface that can be independently assigned colour. 3.1.76 posture: The extent to which the shape of a glyph or set of glyphs appear to incline, including any consequent design or form change. 3.1.77 primary colour stimuli: Three selected coloured lights used to specify the colour of any light presented by the amounts of the three lights that must be mixed additively to produce light matching the light presented. (Any three coloured lights may serve as primaries provided no one of them can be matched by a mixture of the other two. To achieve the maximum gamut of colours by additive mixture, saturated red, green, and blue primaries are commonly used.) 3.1.78 primary colorants: A small number of colorants (dyes or pigments) that may be mixed subtractively to produce a large gamut of colours. The most common primary colorants are cyan (greenish blue), magenta (purplish red), and yellow. 3.1.79 realized edge: The zero-width ideal boundary line of the filled-area if the edge is invisible, and the finite-width displayed line if the edge is visible. 3.1.80 realized interior: In a filled area element, that portion of the ideal interior as extending to and terminating at the realized edge. 3.1.81 region: In the context of closed figures or the POLYGON SET element, an area that is explicitly or implicitly closed, that is a subset of the full area being filled. Regions can be nested, disjoint or overlapping. The boundaries of all regions are considered together when applying the interior test for filling a closed figure or POLYGON SET. 3.1.82 RGB colour space: A colour space with colorimetric coordinates based on red, green and blue reference stimuli or primaries. Colour values may be negative in certain areas outside the gamut defined by the RGB primaries. 3.1.83 RGB-related colour space: A colour space related to the RGB colour space through a linear transformation (3x3 matrix). (Examples are YUV (PAL, SECAM), YIQ (NTSC) or $Y_C R_C B_C$ (CCITT video codes and CCIR601 studio standard).) 3.1.84 reference colour model: Basic colour model within CGM relative to which relationships to specifiable colour models (RGB, CMYK, CIELUV, CIELAB, and RGB-related) are calibrated. The reference colour model is defined by the CIE 1931 standard colorimetric system (XYZ). 3.1.85 saturation: Chromaticness (colourfulness) of an area judged in proportion to its brightness. 3.1.86 segment: A collection of primitives, primitive attributes and some additional attributes associated with the segment as a whole. See "segment attribute". 3.1.87 segment attribute: An attribute associated with a segment as a whole rather than attributes of individual primitives. 3.1.88 size specification mode: A generic term for Line Width Specification Mode, Edge Width Specification Mode, or Marker Size Specification Mode. 3.1.89 skewed: Used to describe stroke precision text when the CHARACTER ORIENTATION vectors are non-perpendicular. CELL ARRAYS when the three defining points form a parallelogram which is not a rectangle; or a segment transformation that causes rectangles to become non-rectangular parallelograms. 3.1.90 symbol: A graphical object which is included by reference at some point in the metafile. 3.1.91 trichromatic system: System for specifying colour stimuli in terms of tristimulus values, based on matching colours by additive mixture of three suitable chosen reference colour stimuli. 3.1.92 view surface: See DISPLAY SURFACE.

ISO/IEC 8632-1:1992 (E)	Definitions
	3.1.54 glyph: A recognizable abstract graphical symbol, which is independent of any specific design. 3.1.55 global segment: A segment that is defined in the Metafile Descriptor (see "segment"), in which case it may be referenced from within any picture. 3.1.56 graphical primitive elements: Metafile elements that describe images in the Metafile. 3.1.57 Graphical Kernel System (GKS): An ISO standard application programmer's interface to graphics (ISO 7942). 3.1.58 graphics device: A device (for example, refresh display, storage tube display, or plotter) on which display images can be represented. 3.1.59 graphic object: A graphical primitive or a compound primitive, together with the associated attributes. 3.1.60 hatch style: A format for filling closed figures. A hatch style consists of one or more sets of lines whose presence represents the interior of the figure in question. 3.1.61 hue: Attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours, red, yellow, green, and blue, or to a combination of two of them. 3.1.62 indexed colour: A colour selection scheme in which the colour index is used to retrieve colour values from a colour table. See COLOUR TABLE, DIRECT COLOUR. 3.1.63 isotropic mapping: A mapping which is invariant with respect to direction; equal scaling in all orthogonal representational dimensions. It is often used to describe the mapping from VDC to distance units on the physical display surface. With isotropic mapping, the angle between any pair of non-parallel line segments remains unchanged; for example, circles remain circles. See "anisotropic mapping". 3.1.64 local segment: A segment which is defined in the picture descriptor or picture body, and whose definition is local to the picture in which it appears. 3.1.65 message: A string of characters used to communicate information to operators at Metafile interpretation time. 3.1.66 metafile: A mechanism for retaining and transporting graphical data and control information. This information contains a device-independent description of one or more pictures. 3.1.67 Metafile Descriptor (MD): A metafile element that describes the format of the metafile (but not its encoding method) and the functionality expected of a metafile interpreter. 3.1.68 metafile element: A functional item that can be used to construct a picture or convey information. 3.1.69 metafile generator: The process or equipment that produces the Computer Graphics Metafile. 3.1.70 metafile interpreter: The process or equipment that reads the Computer Graphics Metafile and interprets the contents. An interpreter may be needed in order to drive a Computer Graphics Interface or other device interface to obtain a picture that resembles the intended picture as closely as possible. 3.1.71 normalized device coordinates (NDC): Coordinates specified in a device-independent coordinate system, normalized to some range (typically 0 to 1). See VDC EXTENT, VDC RANGE, VDC SPACE, VIRTUAL DEVICE COORDINATES. 3.1.72 object clipping: Object clipping is applied to a graphic object. For example, clipping is applied to a line after it has had the width attribute associated with it. 3.1.73 pattern style: A format for filling closed figures with patterns. A pattern style consists of an array of variously coloured or shaded cells. 3.1.74 Picture Descriptor (PD): A set of metafile elements used to set the interpretation modes of attribute elements for the entire picture.

Definitions and abbreviations

Definitions

3.1.93 virtual device: An idealized graphics device that presents a set of graphics capabilities to graphics software or systems via the Computer Graphics Interface.

3.1.94 virtual device coordinates (VDC): The coordinates used to specify position in the VDC space. These are absolute two-dimensional coordinates. See VDC SPACE.

3.1.95 VDC extent: A rectangular region of interest contained within the VDC range. See VDC RANGE, VDC SPACE.

3.1.96 VDC range: A rectangular region within VDC space consisting of the set of all coordinates representable in the declared coordinate type, precision, and encoding format of the metafile. See VDC EXTENT, VDC SPACE.

3.1.97 VDC space: A two-dimensional Cartesian coordinate space of infinite precision and extent. Only a subset of VDC space, the VDC range, is realizable in a metafile. See VDC EXTENT, VDC RANGE, VIRTUAL DEVICE COORDINATES.

3.1.98 weight: The ratio of a glyph's or set of glyphs' stem width to font height.

3.2 Abbreviations

The following abbreviations are used in all parts of ISO/IEC 8632.

ASF	Aspect Source Flag
CDP	Conjugate Diameter Pair
CGI	Computer Graphics Interface
CGM	Computer Graphics Metafile
GDP	Generalized Drawing Primitive
GKS	Graphical Kernel System
MD	Metafile Descriptor
NDC	Normalized Device Coordinate(s)
PD	Picture Descriptor
VDC	Virtual Device Coordinate(s)

4 Concepts

4.1 Introduction

The objective of the Computer Graphics Metafile (CGM) is to provide for the description, storage, and communication of graphical information in a device-independent manner. To accomplish this, ISO/IEC 8632 defines the form (syntax) and functional behaviour (semantics) of a set of elements that may occur in the CGM. The following classes of elements are defined:

- Delimiter Elements, which delimit significant structures within the Metafile.
- Metafile Descriptor Elements, which describe the functional content, default conditions, identification, and characteristics of the CGM.
- Picture Descriptor Elements, which set the interpretation modes of attribute elements for each picture.
- Control Elements, which allow picture boundaries and coordinate representation to be modified.
- Graphical Primitive Elements, which describe the visual components of a picture in the CGM.
- Attribute Elements, which describe the appearance of graphical primitive elements.
- Escape Elements, which describes device- or system-dependent elements used to construct a picture; however, the elements are not otherwise standardized.
- External Elements, which communicate information not directly related to the generation of a graphical image.
- Segment Elements, which enable the grouping and manipulation of elements.

A Computer Graphics Metafile is a collection of elements from this standardized set. The BEGIN METAFILE and END METAFILE elements each occur exactly once in a complete metafile; as many or as few of the elements in the other classes may occur as are needed. A metafile needs to be interpreted in order to display its pictorial content on a graphics device. The descriptor elements give the interpreter sufficient data to interpret metafile elements and to make informed decisions concerning the resources needed for display.

Any CGM contains certain delimiter elements. In addition it may include control elements for metafile interpretation, Picture Descriptor elements for declaring parameter modes of attribute elements and defining the representations of certain attribute index values, graphical primitive elements for defining graphical entities, attribute elements for defining the appearance of the graphical primitive elements, escape elements for accessing non-standardized features of particular devices, and external elements for communication of information external to the definition of the pictures in the CGM. In Version 2 and Version 3 metafiles, graphical output primitives, attributes, and control elements may be grouped in segments.

A minimal correct metafile consists of BEGIN METAFILE, a Metafile Descriptor consisting of METAFILE VERSION and METAFILE ELEMENT LIST, and END METAFILE.

4.2 Delimiter elements

Every metafile starts with a BEGIN METAFILE element and ends with an END METAFILE element. This allows multiple metafiles to be stored or transferred together.

Each picture starts with a BEGIN PICTURE element and ends with an END PICTURE element. Between these delimiters the Picture Descriptor is separated from the picture body by a BEGIN PICTURE BODY element.

Concepts

Delimiter elements

Once the Metafile Descriptor has been read, access to individual pictures, on a random as opposed to sequential basis, may be safely accomplished if the encoding, access mechanism and implementation permit.

BEGIN METAFILE and BEGIN PICTURE both have parameters for a name by which the metafile and picture (respectively) can be identified.

In Version 2 and Version 3 metafiles, primitives may be grouped together to form a composite primitive known as a closed figure. The primitives to be included in the closed figure being defined are delimited by the elements BEGIN FIGURE and END FIGURE.

In Version 2 and Version 3 metafiles, groups of elements, called segments, are delimited by BEGIN SEGMENT and END SEGMENT. Each segment is uniquely identified by a segment identifier. Segments may be defined in the Metafile Descriptor, in the Picture Descriptor, or within picture bodies.

In Version 3 metafiles a compound clipping or shielding region may be defined by line and filled-area elements occurring between BEGIN PROTECTION REGION and END PROTECTION REGION elements.

In Version 3 metafiles a compound path may be defined for drawing a compound line primitive. A compound line is defined by line primitive elements occurring between BEGIN COMPOUND LINE and END COMPOUND LINE elements. A compound path may also be defined for displaying text strings along an arbitrary text path. A compound text path is defined by line primitive elements occurring between BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH elements. The allowed elements and rules for definition are identical for the two types of paths.

In Version 3 metafiles a tile array may be defined by tile array elements occurring between BEGIN TILE ARRAY and END TILE ARRAY.

The exact list of elements which may occur in any of these definition states will be found in the state table, table 8.

4.3 Metafile descriptor elements

The Metafile Descriptor (MD) is a group of elements that describes the functional capabilities required to interpret the CGM. These elements are

METAFILE VERSION	FONT LIST
METAFILE DESCRIPTION	CHARACTER SET LIST
VDC TYPE	CHARACTER CODING ANNOUNCER
INTEGER PRECISION	NAME PRECISION
REAL PRECISION	MAXIMUM VDC EXTENT
INDEX PRECISION	SEGMENT PRIORITY EXTENT
COLOUR PRECISION	COLOUR MODEL
COLOUR INDEX PRECISION	COLOUR CALIBRATION
MAXIMUM COLOUR INDEX	FONT PROPERTIES
COLOUR VALUE EXTENT	GLYPH MAPPING
METAFILE ELEMENT LIST	SYMBOL LIBRARY LIST
METAFILE DEFAULTS REPLACEMENT	

NOTE 1 Other elements, as defined in this part of ISO/IEC 8632, may appear within the Metafile Descriptor within the definition of a global segment.

In a particular metafile, the METAFILE ELEMENT LIST lists at least those standardized elements that occur in the metafile. The CGM interpreter is thus informed of the capabilities required to successfully interpret the Computer Graphics Metafile. The CGM contains a single Metafile Descriptor. The Metafile Descriptor immediately follows the BEGIN METAFILE element in a metafile (with the possible exception

Metafile descriptor elements

of intervening external and escape elements).

METAFILE VERSION and METAFILE ELEMENT LIST shall occur only once in the Metafile Descriptor for metafiles of all three versions.

NOTE 2 It is recommended that the following elements: METAFILE VERSION, METAFILE ELEMENT LIST and (possibly multiple occurrences of) METAFILE DESCRIPTION appear first in the Metafile Descriptor and in the order listed.

4.3.1 Identification

The identifying information includes declaration of the version of ISO/IEC 8632, by the METAFILE VERSION element, and descriptive information about the metafile (origin, owner, generation date, etc.), by the METAFILE DESCRIPTION element.

4.3.2 Functional capability

The contents of the Computer Graphics Metafile are defined by the METAFILE ELEMENT LIST element. This shall contain a list of the non-mandatory elements that are utilized in the metafile (see 4.1 for mandatory elements). Several shorthand names for CGM elements are also provided for use with the METAFILE ELEMENT LIST. The shorthand names shall not be considered macro names, nor shall they be construed to be levels of conformance.

4.3.2.1 Drawing set

The drawing set includes the mandatory CGM elements (i.e., those that shall appear in every conforming CGM) and most of the graphical primitive elements and attribute elements. The drawing set is specified by the shorthand name DRAWING SET.

The elements included in the drawing set are:

BEGIN METAFILE	LINE TYPE
END METAFILE	LINE WIDTH
BEGIN PICTURE	LINE COLOUR
BEGIN PICTURE BODY	MARKER BUNDLE INDEX
END PICTURE	MARKER TYPE
METAFILE VERSION	MARKER SIZE
METAFILE DESCRIPTION	MARKER COLOUR
VDC TYPE	TEXT BUNDLE INDEX
METAFILE ELEMENT LIST	TEXT FONT INDEX
AUXILIARY COLOUR	TEXT PRECISION
TRANSPARENCY	CHARACTER EXPANSION FACTOR
CLIP RECTANGLE	CHARACTER SPACING
CLIP INDICATOR	TEXT COLOUR
VDC EXTENT	CHARACTER HEIGHT
BACKGROUND COLOUR	CHARACTER ORIENTATION
COLOUR SELECTION MODE	TEXT PATH
POLYLINE	TEXT ALIGNMENT
DISJOINT POLYLINE	FILL BUNDLE INDEX
POLYMARKER	INTERIOR STYLE
TEXT	FILL COLOUR
RESTRICTED TEXT	HATCH INDEX

Concepts	Metafile descriptor elements
APPEND TEXT POLYGON POLYGON SET CELL ARRAY GENERALIZED DRAWING PRIMITIVE RECTANGLE CIRCLE CIRCULAR ARC 3 POINT CIRCULAR ARC 3 POINT CLOSE CIRCULAR ARC CENTRE CIRCULAR ARC CENTRE CLOSE ELLIPSE ELLIPTICAL ARC ELLIPTICAL ARC CLOSE LINE BUNDLE INDEX	PATTERN INDEX EDGE BUNDLE INDEX EDGE TYPE EDGE WIDTH EDGE COLOUR EDGE VISIBILITY FILL REFERENCE POINT PATTERN TABLE PATTERN SIZE COLOUR TABLE ASPECT SOURCE FLAGS ESCAPE MESSAGE APPLICATION DATA

4.3.2.2 Drawing plus control set

The drawing-plus-control set may be used to indicate all of the elements in the drawing set plus additional control, Metafile Descriptor, Picture Descriptor, and attribute elements. It is specified by the shorthand name DRAWING PLUS CONTROL SET.

The elements included in the drawing-plus-control set are all of the elements in the drawing set and the following elements:

INTEGER PRECISION REAL PRECISION INDEX PRECISION COLOUR PRECISION COLOUR INDEX PRECISION MAXIMUM COLOUR INDEX COLOUR VALUE EXTENT METAFILE DEFAULTS REPLACEMENT FONT LIST CHARACTER SET LIST	CHARACTER CODING ANNOUNCER VDC INTEGER PRECISION VDC REAL PRECISION SCALING MODE LINE WIDTH SPECIFICATION MODE MARKER SIZE SPECIFICATION MODE EDGE WIDTH SPECIFICATION MODE CHARACTER SET INDEX ALTERNATE CHARACTER SET INDEX
---	---

NOTE — The drawing-plus-control set essentially constitutes a "Version-1 set", however that designation was not defined in the original ISO 8632:1987 and therefore cannot be used for reasons of upward compatibility of versions.

4.3.2.3 Version 2 set

The Version-2 set includes all elements from the drawing-plus-control set and the following elements defined in Version 2 metafiles:

BEGIN SEGMENT END SEGMENT BEGIN FIGURE END FIGURE NAME PRECISION	MARKER CLIPPING MODE EDGE CLIPPING MODE NEW REGION SAVE PRIMITIVE CONTEXT RESTORE PRIMITIVE CONTEXT
--	---

Metafile descriptor elements	Concepts
MAXIMUM VDC EXTENT SEGMENT PRIORITY EXTENT DEVICE VIEWPORT DEVICE VIEWPORT MAPPING DEVICE VIEWPORT SPECIFICATION MODE LINE REPRESENTATION MARKER REPRESENTATION TEXT REPRESENTATION FILL REPRESENTATION EDGE REPRESENTATION LINE CLIPPING MODE	CIRCULAR ARC CENTRE REVERSED CONNECTING EDGE PICK IDENTIFIER COPY SEGMENT INHERITANCE FILTER CLIP INHERITANCE SEGMENT TRANSFORMATION SEGMENT HIGHLIGHTING SEGMENT DISPLAY PRIORITY SEGMENT PICK PRIORITY
4.3.2.4 Extended primitives set	
The extended-primitives set may be used to indicate those primitives which are available in Version 1 metafiles but are not defined in ISO 7942 (GKS). These elements are:	
DISJOINT POLYLINE RESTRICTED TEXT APPEND TEXT POLYGON SET RECTANGLE CIRCLE CIRCULAR ARC 3 POINT CIRCULAR ARC 3 POINT CLOSE	CIRCULAR ARC CENTRE CIRCULAR ARC CENTRE CLOSE CIRCULAR ARC CENTRE REVERSED ELLIPSE ELLIPTICAL ARC ELLIPTICAL ARC CLOSE CONNECTING EDGE

4.3.2.5 Version 2 GKSM set

The Version-2-GKSM set is a set of Version 2 metafile elements for ISO 7942 (GKS) picture capture. The elements included in the Version-2-GKSM set are:

BEGIN METAFILE BEGIN PICTURE BEGIN PICTURE BODY END PICTURE BEGIN SEGMENT END SEGMENT END METAFILE METAFILE VERSION METAFILE DESCRIPTION VDC TYPE INTEGER PRECISION REAL PRECISION INDEX PRECISION COLOUR PRECISION COLOUR INDEX PRECISION NAME PRECISION MAXIMUM COLOUR INDEX COLOUR VALUE EXTENT METAFILE ELEMENT LIST	CELL ARRAY GDP LINE BUNDLE INDEX LINE TYPE LINE WIDTH LINE COLOUR MARKER BUNDLE INDEX MARKER TYPE MARKER SIZE MARKER COLOUR TEXT BUNDLE INDEX TEXT FONT INDEX TEXT PRECISION CHARACTER EXPANSION FACTOR CHARACTER SPACING TEXT COLOUR CHARACTER HEIGHT CHARACTER ORIENTATION TEXT PATH
--	--

METAFILE DEFAULTS REPLACEMENT
FONT LIST
CHARACTER SET LIST
CHARACTER CODING ANNOUNCER
MAXIMUM VDC EXTENT
SEGMENT PRIORITY EXTENT
VDC EXTENT
DEVICE VIEWPORT
DEVICE VIEWPORT MAPPING
DEVICE VIEWPORT SPECIFICATION MODE
LINE REPRESENTATION
MARKER REPRESENTATION
TEXT REPRESENTATION
FILL REPRESENTATION
VDC INTEGER PRECISION
VDC REAL PRECISION
CLIP RECT ANGLE
POLYLINE
POLYMARKER
TEXT
POLYGON

TEXT ALIGNMENT
CHARACTER SET INDEX
ALTERNATE CHARACTER SET INDEX
FILL BUNDLE INDEX
INTERIOR STYLE
FILL COLOUR
HATCH INDEX
PATTERN INDEX
FILL REFERENCE POINT
PATTERN TABLE
PATTERN SIZE
COLOUR TABLE
ASPECT SOURCE FLAGS
PICK IDENTIFIER
ESCAPE
MESSAGE
APPLICATION DATA
SEGMENT TRANSFORMATION
SEGMENT HIGHLIGHTING
SEGMENT DISPLAY PRIORITY
SEGMENT PICK PRIORITY

4.3.2.6 Version 3 set

The Version-3 set may be used to indicate all elements in the Version-2 set and the elements:

BEGIN PROTECTION REGION
END PROTECTION REGION
BEGIN COMPOUND LINE
END COMPOUND LINE
BEGIN COMPOUND TEXT PATH
END COMPOUND TEXT PATH
BEGIN TILE ARRAY
END TILE ARRAY
COLOUR MODEL
COLOUR CALIBRATION
FONT PROPERTIES
GLYPH MAPPING
SYMBOL LIBRARY LIST
INTERIOR STYLE SPECIFICATION MODE
LINE AND EDGE TYPE DEFINITION
HATCH STYLE DEFINITION
GEOMETRIC PATTERN DEFINITION
PROTECTION REGION INDICATOR
GENERALIZED TEXT PATH MODE
MITRE LIMIT
TRANSPARENT CELL COLOUR
HYPERBOLIC ARC

PARABOLIC ARC
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER
POLYSYMBOL
BITONAL TILE
TILE
LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET
TEXT SCORE TYPE
RESTRICTED TEXT TYPE
INTERPOLATED INTERIOR
EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET
SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL SIZE
SYMBOL ORIENTATION

4.3.3 Default metafile state

The default state is the state to which the interpreter is returned at the start of each picture. The default states of all metafile elements are defined in clause 6. These default values may be selectively replaced by using the METAFILE DEFAULTS REPLACEMENT element. The correspondence between character set indexes and registered or private character sets, and the meaning assigned to text font indexes, are also established in the Metafile Descriptor.

4.3.4 Fonts and character sets

4.3.4.1 Font list and font resources

ISO/IEC 9541-1 defines an architecture for font resources; but does not define or standardize applications' use of the information in a font resource — ranging from gross or aggregate properties such as font posture to very specific and detailed properties such as individual glyph metrics. A metafile generator (with its associated application) will be a user of such font resource information. The application, in defining a picture which contains text strings, has knowledge of the properties of the font resource. It makes use of these properties to format or layout strings of text so that the complete strings have the desired characteristics.

CGM is used to transmit such pictures from a generating application to an interpreting application, possibly remote in time and space and possibly of very different architecture and resource availability. The font facilities of CGM are designed to provide a font referencing mechanism. Font referencing is the process of identifying or characterizing a font resource. Referencing may include identification of a specific font by name, or provide sufficient descriptive information to permit identification of a suitable font or substitute. This concept is described in ISO/IEC 9541-1, annex B.

The FONT LIST element of CGM allows the exact naming of a font resource. Such font resources may in the future be registered and given structured names under the mechanisms of ISO/IEC 9541. In the ideal case the metafile interpreter recognizes and has available the font resource named in the FONT LIST. For cases where the named font is not available to the interpreter, the CGM has elements (FONT PROPERTIES and GLYPH MAPPING) which allow generators to pass to interpreters additional descriptive information about desired fonts and font resources. An alternative font can be selected by an interpreter through this descriptive information if the specified one is not available.

4.3.4.2 Font properties

The FONT PROPERTIES element can be used to guide selection of a best fit font if an exact match is not available on a specific device. The font properties which may appear are those in the Minimum Font Description Subset of ISO/IEC 9541-2. Applications may use registered extensions to access additional properties from amongst the ISO/IEC 9541 font properties.

NOTE — Registration of additional font properties is done using the registration procedures of the ISO International Register of Graphical Items, as described in ISO/IEC TR9973.

The element allows the importance of each property to be assigned a priority relative to the other properties. In the case that a font named in the FONT LIST is not available, the priorities of the properties instruct the interpreter of the relative importance of the various characteristics of the requested font. In some cases it may not even be necessary to get a particular font, but rather any font with certain characteristics — boldness, presence of serif, etc. The FONT PROPERTIES element enables generators to specify such concepts. The use of this information by interpreters is not standardized.

4.3.4.3 Character set repertoire in graphical text strings

The CHARACTER SET LIST element allows designation of registered character sets for use in the metafile. It also allows designation of private character sets, and in combination with the GLYPH MAPPING element allows designation of collections of glyphs which are registered (according to the procedures of ISO/IEC 10036).

Concepts

Format effector control characters (NUL, CR, LF, BS, HT, VT, and FF) are permitted in a parameters of type String but their interpretation is implementation dependent. Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER. The other control codes from the C0 and C1 sets are reserved for future standardization.

NOTE — SI and SO are only defined and usable with 7-bit coding.

The mechanisms described in this section for character set definition, and the mechanisms for accessing the character sets (CHARACTER CODING ANNOUNCER, CHARACTER SET INDEX, and ALTERNATE CHARACTER SET INDEX) apply only to string parameters (S) of graphical text elements (TEXT, RES-TRICTED TEXT, APPEND TEXT, and appropriate GDPs). They do not apply to non-graphical text strings (SF) such as the *metafile identifier* parameter of the BEGIN METAFILE element.

For designation of ISO registered character sets, two pieces of information are specified by the parameters: a character set type, and the tail of its designating sequence by which it is known in the ISO register.

There are five types of character sets: 94-character G-sets, 96-character G-sets, 94-character multibyte G-sets, 96-character multibyte G-sets, and character sets intended to be designated as "complete codes".

94-CHARACTER G-SETS. These character sets are designated by ISO 2022 escape sequences of the form <ESC> <I> <O> <F>. Here, <I> is either 2/8, 2/9, 2/10, or 2/11; <O> represents zero or more intermediate characters from column 2 of the code chart; and <F> is a final character from columns 3 through 7 of the code chart. If <F> is from column 3 of the code chart, the character set is a "private" character set. If <F> is from columns 4 through 7 of the code chart, the character set is a "standard" character set in the sense that it and its designating escape sequences are registered in the International Register Of Coded Character Sets To Be Used With Escape Sequences.

For 94-character G-sets, the character set declaration consists of '94-character G-set', followed by a string consisting of all characters in the ISO 2022 designating escape sequence except the first two characters, <ESC> <I>.

For example, the G-set from the U.K.'s national 7-bit character set is registered in the International Register Of Coded Character Sets To Be Used With Escape Sequences. Its designating escape sequences are as follows:

```
<ESC> 2/8 4/1 (to designate it as G0)
<ESC> 2/9 4/1 (to designate it as G1)
<ESC> 2/10 4/1 (to designate it as G2)
<ESC> 2/11 4/1 (to designate it as G3)
```

Again, the French character set (1982 version, from the 1982 version of AFNOR NF Z 62-010) is registered in the International Register Of Coded Character Sets To Be Used With Escape Sequences. Its designating escape sequences are as follows:

```
<ESC> 2/8 6/6 (to designate it as G0)
<ESC> 2/9 6/6 (to designate it as G1)
<ESC> 2/10 6/6 (to designate it as G2)
<ESC> 2/11 6/6 (to designate it as G3)
```

Therefore, a CHARACTER SET LIST element could specify that the U.K. character set is to be referred to by character set index 1, and the French character set by character set index 2, as follows:

Metafile descriptor elements

<CHARACTER-SET-LIST: U.K., French>

```
'94-character G-set' 4/1
'94-character G-set' 6/6
```

96-CHARACTER G-SETS. These character sets are similar to 94-character G-sets, but include the code positions 2/0 and 7/15, which are excluded from 94-character G-sets. Their ISO 2022 designating escape sequences take the form <ESC> <I> <O> <F>, where the first intermediate character <I> is either 2/13, 2/14, or 2/15. The remainder of the escape sequence is similar to the escape sequences for 94-character G-sets: zero or more intermediate characters from column 2 of the code chart and a final character from columns 3 through 7 of the code chart.

For 96-character G-sets, the character set declaration consists of '96-character G-set', followed by a string consisting of all characters in the ISO 2022 designating escape sequence except the first two characters, <ESC> <I>.

It is possible for interchanging parties to agree on a private 96-character G-set whose designating escape sequences would end with a character from column 3 of the code chart. For example, the following might be private escape sequences to designate such a G-set:

```
<ESC> 2/13 3/0 (to designate it as G1)
<ESC> 2/14 3/0 (to designate it as G2)
<ESC> 2/15 3/0 (to designate it as G3)
```

(96-character G-sets may not be designated as G0 sets.)

For example, the following CHARACTER SET LIST element establishes the U.K. 94-character G-set, the French 94-character G-set, and a private 96-character G-set as the character sets named by character set indexes 1, 2, and 3, respectively:

<CHARACTER-SET-LIST: U.K., French, private 96-character G-set>

```
'94-character G-set' 4/1
'94-character G-set' 6/6
'96-character G-set' 3/0
```

94-CHARACTER MULTIBYTE G-SETS. A 94-character multibyte G-set can contain 94 to the Nth power characters, each coded as a sequence of N bytes from columns 2 through 7 of the code chart, not including the bytes 2/0 and 7/15, which are excluded from 94-character G-sets. For example, a 94-character 2-byte G-set can contain 8,836 characters.

The ISO 2022 designating escape sequences for 94-character multibyte G-sets have the following forms:

```
<ESC> 2/4 2/8 <F> (to designate it as G0)
<ESC> 2/4 2/9 <F> (to designate it as G1)
<ESC> 2/4 2/10 <F> (to designate it as G2)
<ESC> 2/4 2/11 <F> (to designate it as G3)
```

For compatibility with the first version of ISO 2022, when <I> is 4/0, 4/1, or 4/2 the byte 2/8 may be left out of the designating escape sequence as a GO set.

For 94-character multibyte G-sets, the character set declaration consists of '94-character multibyte G-set', followed by a string consisting only of the final character in the ISO 2022 designating escape sequence.

For example, a Japanese 2-byte character set of 6802 graphic characters has been registered in the International Register Of Coded Character Sets To Be Used With Escape Sequences, and its designating escape sequences have the form shown above, with the final character <F> being 4/0. Thus, the following CHARACTER SET LIST element could be used to specify that this 2-byte Japanese character set is to be referred to by character set index 1:

<CHARACTER-SET-LIST: Japanese 2-byte character set>

 '94-character multibyte G-set' 4/0

96-CHARACTER MULTIBYTE G-SETS. A 96-character multibyte G-set is similar to a 94-character multibyte G-set except that it can include the bytes 2/0 and 7/15. Thus, a 96-character 2-byte G-set could have 96 times 96 (or 9216) 2-byte character codes.

The ISO 2022 designating escape sequences for 96-character multibyte G-sets have the following forms:

<ESC> 2/4 2/13 <F> (to designate it as G1)
<ESC> 2/4 2/14 <F> (to designate it as G2)
<ESC> 2/4 2/15 <F> (to designate it as G3)

It is not possible to designate a 96-character multibyte G-set as a G0 set.

The character set declaration for a 96-character multibyte G-set consists of '96-character multibyte G-set' followed by a string consisting only of the final character <F> in the character set's ISO 2022 designating escape sequence.

At the time of this publication, no 96-character multibyte G-sets have been registered in the International Register of Coded Character Sets To Be Used with Escape Sequences.

CHARACTER SETS INTENDED TO BE DESIGNATED AS COMPLETE CODES. Other character sets may not fit the ISO 2022 "G-set" structure. ISO 2022 provides an escape sequence format for invoking coding systems different from ISO 2022. The complete code escape sequences have the following form

<ESC> 2/5 <I> <F>

where <I> means "zero or more characters from column 2 of the code chart", and <F> is a final character from columns 3 through 7 of the code chart. If <F> is from column 3, the coding system is a private code. If <F> is from columns 4 through 7, it is a code for which a designating and invoking escape sequence has been registered in the International Register Of Coded Character Sets To Be Used With Escape Sequences.

The character set declaration for a character set that would be invoked as a coding system different from ISO 2022 consists of 'complete code' followed by a string consisting only of those characters in the code's ISO 2022 escape sequence which come after the first two characters, <ESC> 2/5.

As well as using a registered complete code, a private code could be used. For example, suppose the interchanging parties have agreed on a private 8-bit code to be invoked by the following escape sequence:

<ESC> 2/5 3/0

The following CHARACTER SET LIST element would declare the French character set to have character set index 1 and that 8-bit private code to have character set index 2:

<CHARACTER-SET-LIST: French, private coding system>

 '94-character G-set' 6/6

 'complete code' 3/0

Information regarding the designation sequence tail parameter is found in the International Register of Coded Character Sets to be Used with Escape Sequences. This register is maintained by the Registration

Authority for ISO 2375, which is the European Computer Manufacturers Association (ECMA), Rue du Rhone 114, CH-1204, Geneva, Switzerland.

4.3.4.4 Glyph repertoire

ISO/IEC 10036 specifies a procedure and a registrar (registering authority) for registering typographic glyph collections. There currently is no standard that associates codes (i.e., character codes) with these glyphs. However the registrar — the Association for Font Information Interchange, or AFI — assigns a unique 4-byte integer identifier with each glyph.

This part of ISO/IEC 8632 defines a means to access these registered glyph collections for use in graphical text strings. The GLYPH MAPPING element associates the AFI 4-byte identifiers with single-byte or multi-byte codes. A set of such codes is defined as a collection, forming a locally defined character set for use within the metafile. The local character set is associated with an index, and within the body of the CGM the normal character set access and switching mechanisms (based upon, and adapted from, ISO 2022) may be used to access the AFI registered glyphs within CGM text strings.

NOTE — The glyph complement is a property of a font resource in the ISO/IEC 9541 font architecture. When the separate mechanisms of ISO/IEC 8632 for font reference and glyph access are used there is potential for incompatibility between the specifications — the requested glyph complement may not be representable in the requested font.

4.3.4.5 Character set selection in non-graphical text strings

The mechanisms discussed in the preceding sections apply to graphical text strings — text strings which result in display as graphical entities. These are the parameters of TEXT, RESTRICTED TEXT, APPEND TEXT, and possibly GDP elements.

They do not apply to other, non-graphical text strings which occur as metafile parameters. See 5.1, type String Fixed (SF). For these strings, character set selection shall be accomplished only through use of embedded ISO 2022 controls, to switch sets within the string.

For the purposes of coding type SF parameters in ISO/IEC 8632 metafiles, 7-bit coding environment shall be used. It is not possible, either using ISO 2022 controls or any other defined facilities in this International Standard, to select other than 7-bit coding for type SF parameters.

In SF parameters, therefore, the two character sets which are currently designated as G0 and G1 may be accessed by using: SI to invoke G1 into positions 2/1 to 7/14 of the 7-bit code chart; and using SO to invoke G0 into positions 2/1 to 7/14 of the 7-bit code chart.

BEGIN METAFILE causes ISO 646 International Reference Version to be designated as the G0 set and ISO 8859-1, Right Hand Side of Latin Alphabet Nr. 1 to be designated as the G1 set. In other words, ISO 646 is the default character set of type SF parameters, an SI embedded in the string will cause subsequent 7-bit codes to access the characters of ISO 8859-1, and an SO embedded in the string will cause subsequent 7-bit codes to access ISO 646 again. These two character set may be switched between using only SI and SO — no ISO 2022 designation sequences are required.

EXAMPLE — Many of the characters of the Latin-based alphabets would be accessible with ISO 646 and the Right Hand Side of ISO 8859-1, Latin Alphabet Number 1. These are the default G0 and G1 sets of the metafile. ISO 2022 sequences could be used to designate any other character sets as G0 and/or G1, and then SI/SO could be used to switch between these sets.

The ISO 2022 designating sequence

ESC 2/8 4/2,

for example, would designate ISO 646, U.S. National Character Set (ASCII) as the G0 set (the default set).

Similarly, the ISO 2022 designating sequence

ESC 2/13 4/2

would designate ASCII as the G1 set, superseding the default G1 set (ISO 8859-1) — character codes following an SI would then refer to ASCII, not ISO 8859-1.

Format effector control characters (NUL, CR, LF, BS, HT, VT, and FF) are permitted in a parameters of type String Fixed but their interpretation is implementation dependent. Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER. The other control codes from the C0 and C1 sets are reserved for future standardization.

4.4 Picture descriptor elements

Picture Descriptor elements include elements to declare the parameter modes of other elements for an entire picture, to configure that portion of coordinate space that is of interest in the picture, and to set the colour to which the view surface is cleared at the start of the picture. These elements are:

SCALING MODE
COLOUR SELECTION MODE
LINE WIDTH SPECIFICATION MODE
MARKER SIZE SPECIFICATION MODE
EDGE WIDTH SPECIFICATION MODE
VDC EXTENT
BACKGROUND COLOUR
DEVICE VIEWPORT
DEVICE VIEWPORT SPECIFICATION MODE
DEVICE VIEWPORT MAPPING
LINE REPRESENTATION
MARKER REPRESENTATION
TEXT REPRESENTATION
FILL REPRESENTATION
EDGE REPRESENTATION
INTERIOR STYLE SPECIFICATION MODE
LINE AND EDGE TYPE DEFINITION
HATCH STYLE DEFINITION
GEOMETRIC PATTERN DEFINITION

In Version 1 metafiles, any Picture Descriptor elements within a picture shall appear after the BEGIN PICTURE element and before the BEGIN PICTURE BODY element. Escape and external elements are permitted in the Picture Descriptor.

In Version 2 and Version 3 metafiles, some of the Picture Descriptor elements may appear within the picture body. Table 8 indicates which Picture Descriptor elements may appear and where they may appear for each metafile version.

4.4.1 Scaling mode

VDC space may be either an abstract space, which may be mapped to an arbitrary size on a physical device, or a metric space, which is intended to be mapped to a particular size. Selection of the mode to be used can be made on a picture-by-picture basis by means of the SCALING MODE element. The scaling mode element provides a flag to select abstract space or metric space, and a scale factor which specifies the number of millimeters per VDC unit when metric space is selected.

4.4.2 Colour selection mode

COLOUR SELECTION MODE selects either indexed or direct colour specification and is described further under colour attributes, 4.7.6. For Version 1 metafiles, the selection is for the whole picture.

4.4.3 Specification modes

Line width, marker size, interiors of filled-area elements, and edge width may be specified in more than one way. The width of lines, for example, may be specified as either a measure in VDC units, a scale factor to be applied to a device-dependent nominal line width at interpretation time, a fraction of the device view surface, or a measure in millimetres. For each attribute element having such multiple modes, there is an associated control element that defines the mode of the parameter of the attribute element.

4.4.4 VDC extent

There is a metafile element to define the VDC extent. The extent is set with the VDC EXTENT element by specifying the addresses (in VDC) of the lower-left corner and the upper-right corner of this extent as seen by the viewer of the picture. Specification of values outside the VDC extent is permitted in CGM elements. It is intended that the visible portion of an image be contained within the VDC extent. It thus provides a frame for the region of interest in a picture. The values of the coordinates for either dimension may be either increasing or decreasing from the lower-left to the upper-right corner. For example, for devices with an upper-left origin, a picture may be described in coordinates that map directly to the device but still may be displayed correctly on a device with a lower-left origin. figure 1 illustrates these concepts.

The VDC extent thus establishes the sense and orientation of VDC space (that is, the directions of the positive x (+x) and positive y (+y) axes, and whether the +y axis is 90° clockwise or 90° counterclockwise from the +x axis). In particular, VDC EXTENT establishes the direction of positive and negative angles as follows: positive 90° is defined to be the right angle from the positive x-axis to the positive y-axis (see figure 1).

Note that some attributes such as text attributes (for example, the directions of the 'up' and 'base' component vectors of CHARACTER ORIENTATION, and therefore the meaning of the enumerative values 'right', 'left', 'up', 'down') are intimately bound to these definitions.

The default state of the extent is specified in clause 6, and can be changed in the METAFILE DEFAULTS REPLACEMENT element in the Metafile Descriptor. VDC EXTENT returns to this default state at the beginning of each picture. MAXIMUM VDC EXTENT defines an extent which bounds the VDC extent values which may be found in the metafile. It may be, but need not be, a closest bound in the sense that it exactly equals the union of the extent rectangles in the metafile. This element may be used, for example, to map integer virtual device coordinates of the metafile to a unit square in a normalized device space.

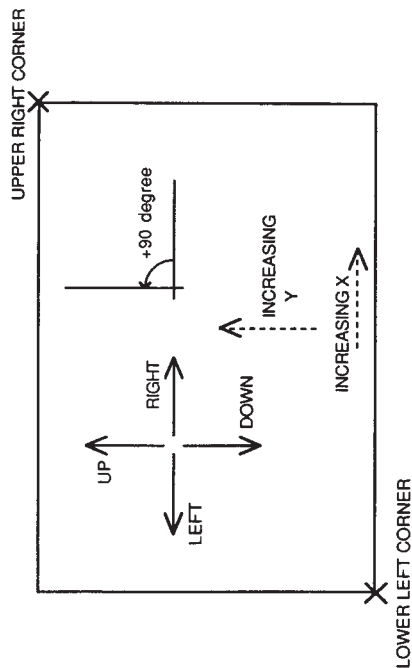
4.4.5 CGM tailoring

The ability to specify the VDC range and the VDC extent provides the flexibility to configure the metafile addressability in any way desired. It can be configured as an abstract, normalized address range for maximum device independence. It can also be configured to mimic the addressability of a particular target device in order to take advantage of particular device characteristics. The address range of such a device-specific metafile is just another normalized address range with the normalization limits inherent in the VDC-customizing element; therefore, device independence is maintained.

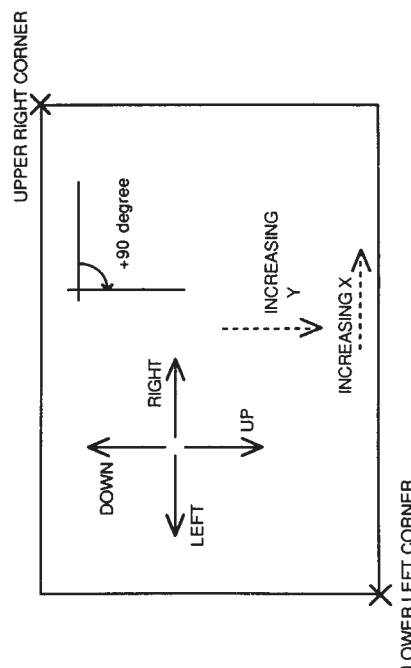
Such tailoring of the coordinates in a metafile can eliminate the need for transformation of coordinates at metafile interpretation time for the target device. The ability to specify the VDC extent thus allows for the exact registration of coordinates in a metafile with addressable points on the target graphics device.

The use of VDC EXTENT to directly encode world coordinates of large dynamic range and very small granularity will likely result in performance penalties at metafile interpretation time, and may result in decreased portability if such VDC extents exceed those compatible with less capable metafile interpreters.

In addition to VDC tailoring, a metafile generator can limit or tailor the functional content of a metafile to accommodate particular devices or applications, and announce such functional tailoring through the use of METAFILE ELEMENT LIST.



VDC EXTENT 0.0,0.0,1.0,0.75



VDC EXTENT 0.0,0.8,5,11.0,0.0

Figure 1 — VDC EXTENT establishes the direction of positive and negative angles

4.4.6 Background colour

Each picture defines a graphical image that is independent of the other images in a metafile. The background colour of the image may be specified by the BACKGROUND COLOUR Picture Descriptor element. If that element is not contained in the Picture Descriptor, the background colour of the image is the default background colour, whether that default is as specified in clause 6, or has been specified in the METAFILE DEFAULTS REPLACEMENT element.

The single parameter of BACKGROUND COLOUR is always a direct colour, regardless of the current value of COLOUR SELECTION MODE. If the COLOUR SELECTION MODE is indexed, then the BACKGROUND COLOUR element defines the initial representation of colour index 0 for the picture.

4.4.7 Device viewport control

The device viewport specifies the region of the device display surface into which the VDC extent is to be mapped on interpretation. VDC-to-Device mapping is determined by the VDC extent, device viewport, and device viewport mapping.

The position of the device viewport is specified in one of three coordinate systems selected by the DEVICE VIEWPORT SPECIFICATION MODE element:

- by fraction [0.0 to 1.0] of the available display surface, which allows reasonable placement and relative sizing of the viewport;
- in millimetres times a scale factor, which allows absolute sizing of images;
- in physical device coordinates.

The device viewport is specified in terms of two points on the device display surface at diagonally opposite corners of the rectangle. Mirroring or 180° rotation of the image may be achieved by specifying the corners in some way other than the first as below and to the left of the second.

The DEVICE VIEWPORT MAPPING element may be used to force isotropic mapping even if the specified VDC extent and device viewport would not otherwise have led to one. In such a case, the VDC extent is mapped on to a subset of the specified device viewport on interpretation. This subset is defined by shrinking either the vertical or horizontal dimension of the specified viewport as needed to reach the required aspect ratio. This smaller "effective viewport" is then used to define the coordinate mapping from VDC to the device's coordinates. The placement of the effective viewport rectangle within the original one can be specified. This placement can be one of 'left', 'right' or 'centred' when the shrinking is horizontal, and 'top', 'bottom' or 'centred' when it is vertical. These meanings are relative to the display surface of the device.

The VDC-to-Device mapping maps the first point specifying the VDC extent on to the corner of the effective viewport corresponding to the first point specifying the device viewport, and similarly for the second point. The mapping is linear in each dimension, but is not necessarily isotropic (for example, a circle in VDC may not appear as a circle to the viewer).

Both the way VDC space is oriented relative to the display surface and the way the effective viewport is placed on the physical device may lead to mirroring and 180° rotation.

The behaviour of primitives and attributes with significance in VDC space under transformations is further described in 4.6.

If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appears then the default values for device viewport take precedence.

4.4.8 Representations

The elements LINE REPRESENTATION, MARKER REPRESENTATION, TEXT REPRESENTATION, FILL REPRESENTATION and EDGE REPRESENTATION are used to set all of the attribute values in a bundle table entry at the same time. The attributes that may be bundled are described in 4.7.

4.4.9 Definable attributes

A precise definition of line and edge types as well as hatch styles may be accomplished as described in 4.7. Moreover, geometric patterns may be defined (in addition to patterns which are arrays of colours confined to parallelogram-like cells) using the segment definition mechanism. See 4.7.

4.5 Control elements

Control elements specify address space, clipping boundaries, and format descriptions of the CGM elements. Control of some of these format descriptions may be accomplished by Metafile Descriptor elements, while control of others is accomplished by control elements, which may appear in the picture bodies in the metafile. Those items in the former category are fixed for a given metafile, while those in the latter category are changeable; that is, they may change within a picture. Some of the control elements may appear in the Picture Descriptor if this is permitted by the formal grammar for the metafile version. See table 8.

4.5.1 VDC space and range

The graphical primitive elements of a metafile define virtual images. The coordinates of these elements (that is, the addresses of points in the virtual image) are absolute two-dimensional Virtual Device Coordinates (VDC). VDC space is a two-dimensional coordinate space of infinite precision and infinite extent. Only a subset of VDC space, the VDC range, is realizable. The VDC range comprises all coordinates representable in the format specified by the declared VDC TYPE and (depending on the type) the VDC INTEGER PRECISION or VDC REAL PRECISION.

The VDC range cannot be set directly; it is completely determined by VDC TYPE and either VDC INTEGER PRECISION or VDC REAL PRECISION elements in the metafile. These elements are controllable, some by dynamic elements in the metafile body and some by static elements in the Metafile Descriptor. Note that the VDC range thus defined (a rectangular subregion of the VDC space) does not enclose a continuum of values, but has a distinct granularity. Regardless of the aspect ratio of the VDC range and the granularity within the range, it is implicit that one VDC unit in the x-direction represents the same distance as one VDC unit in the y-direction in VDC space.

4.5.2 Clipping

In order to defer clipping of graphical primitive elements (particularly, expandable elements such as CIRCULAR ARC 3 POINT, TEXT, etc.) until metafile interpretation time, a clipping control feature is provided in the CGM. Clipping control is achieved by defining CLIP RECTANGLE in VDC space. Whether clipping to the limits of CLIP RECTANGLE actually occurs at metafile interpretation time is controlled by the CLIP INDICATOR element that sets the mode of the metafile to 'on' or 'off'.

In Version 3 metafiles, primitives may also be clipped against more general regions as defined by BEGIN PROTECTION REGION and END PROTECTION REGION, and as controlled by PROTECTION REGION INDICATOR (see 4.5.4). Clipping effects in the remainder of this section are described in terms of the basic CLIP RECTANGLE capability, but they apply equally to general clip regions.

There are three different clipping modes for lines, markers and edges. The required clipping mode is recorded in the metafile with the elements: LINE CLIPPING MODE, MARKER CLIPPING MODE, and

EDGE CLIPPING MODE. When the CLIP INDICATOR associated with a graphical primitive is 'on', only those parts of a graphical primitive that are considered inside the effective clipping region are rendered on interpretation. The object clipping modes allow precise specification as to how clipping is applied to primitives on interpretation.

Clipping may be either 'locus' or 'shape' or 'locus then shape'. Conceptually, a locus is a mathematical object like a point or line segment, while a shape is an area in 2-dimensional space. Loci are 0-, 1- or 2-dimensional subsets of real-valued 2-space. For markers and text, they are points. For lines, they are the individual line segments or portions of arcs. The locus of an area is the shape and the boundary. Shapes reflect the realization of geometric attributes and are generally 2-dimensional subsets of real-valued 2-space.

'Locus' clipping is applied for each portion of a graphical object based on its mathematical location and is independent of the area it will occupy after rendering. For example, no portion of a line segment is rendered if the ideal mathematical line lies outside the effective clipping region (even if its line width would carry some portion of the rendering of it into the clipping rectangle); no portion of a marker is rendered if its location lies outside the clipping rectangle.

If 'locus' clipping is used, the rendering is applied to the locus of the graphical object after clipping. The resulting rendered shape areas may therefore extend outside the effective clipping region.

'Shape' clipping is applied after the abstract rendering of shape in device coordinate space. The 2-dimensional point set associated with the graphical object is intersected with the effective clipping region, which has been transformed to device coordinate space.

'Locus then shape' clipping allows the specification that both 'locus' and 'shape' clipping be applied to graphical objects as described above. In this case however, the rendered shape will not extend outside the effective clipping region. A thick line whose locus is outside the clip rectangle will not have any portion visible even if its line width would carry some portion of the rendering inside the clip rectangle.

Figure 2 shows some examples of the effect of the clipping modes.

When a width or size specification mode is 'scaled', 'fractional', or 'mm', the rendering of shape proceeds in device coordinate space after application of the VDC-to-Device mapping.

When a width or size specification mode is 'absolute', the rendering of shape proceeds, in VDC space before application of the copy transformation and before application of the segment transformation (if the primitive is in a copied segment), and before the VDC-to-Device mapping.

Fill and text primitives do not have associated object clipping modes (though the edge of a fill primitive and the boundary edges of a closed figure do). Clipping for fill primitives is always consistent with 'shape' clipping (see 4.6.4). For text primitives, the type of clipping is determined by the associated text precision:

- For 'string' precision text, clipping proceeds, on a per string basis, in a manner consistent with 'locus' clipping.
- For 'character' precision text, clipping proceeds, on a per character basis, in a manner consistent with 'locus' clipping.
- For 'stroke' precision text, the clipping always proceeds in a manner consistent with 'shape' clipping.

NOTE — It is valid for an interpreter to perform 'shape' clipping of text in all cases, i.e., 'stroke' text precision is a valid realization of 'character' and 'string' text precision (and 'character' text precision is a valid realization of 'string' text precision).

Clip rectangles applied to graphical primitive elements within segments may be subject to transformations in VDC space. Intersection of clip rectangles (untransformed or transformed) may result in polygonal clipping boundaries (see 4.10.5).

Concepts

4.5.3 Save and restore primitive context

Two elements, **SAVE PRIMITIVE CONTEXT** and **RESTORE PRIMITIVE CONTEXT**, are provided to save and restore a context; that is, attributes and control elements as collections. This capability allows a list of attributes and control elements (see 5.5.11) to be stored in the metafile which can be referenced by name at a later point in the metafile. This capability can be used to save and restore attributes and control elements in conjunction with opening and closing segments.

The values for attributes controlled by specification or selection modes are saved in the mode in which they were last specified along with the value of the corresponding mode. In restoring a context, the current specification and selection modes are not changed.

4.5.4 Compound clipping and shielding

The clipping and shielding elements consist of **BEGIN PROTECTION REGION**, **END PROTECTION REGION**, and **PROTECTION REGION INDICATOR**. The **BEGIN PROTECTION REGION** and **END PROTECTION REGION** elements are delimiter elements, and the **PROTECTION REGION INDICATOR** element is a control element.

Protection regions are identified by an index. Protection regions are constructed by the same primitive elements as closed figures. The interior of a given protection region is defined in the same way as the interior of a closed figure. Regions which are constructed by line elements are closed by **NEW REGION**, **END PROTECTION REGION**, or any filled area element. If the endpoints and beginning points of subsequent line elements are not identical they are implicitly connected by a straight line.

If a protection region is used within a segment (i.e., the **PROTECTION REGION INDICATOR** has the value 'clip' or 'shield' for that region inside the segment), it behaves as do clip rectangles with respect to transformations — it transforms by the copy transformation which is associated with the **COPY SEGMENT** element. Protection regions used in segments are also affected by the segment transformations of the segment.

The effective protection region for clipping and/or shielding, when protection regions are defined, or referenced, within segments, is determined by **CLIP INHERITANCE** in the same way as for the simpler rectangle clipping (see 4.10.5).

NOTE — Inheritance of regions is complicated by the fact that, conceptually, the entire list of regions together with their indicators must be subject to the inheritance mechanism, down to the point in the segment hierarchy where reference to a region is made or the region must be applied for clipping or shielding (in the case of the 'intersection' value of **CLIP INHERITANCE**).

If a protection region is defined within a segment, then its definition persists after the **END SEGMENT** terminating that segment definition. This is exactly as for clip rectangles, and primitive attributes. Thus the definition and activation of a region within a local segment in a picture body potentially has effect in the picture body following the segment. If a segment is copied into the picture body with a **COPY SEGMENT** function, then region definitions within the segment do not affect primitives following, again exactly as clip rectangle.

A protection region is associated with an index at definition time. If a region is defined with an index that is already in use, then the old definition is deleted. The associated protection region indicator for the newly defined region takes the initial value 'off'.

When a protection region is set for clipping, only the portions of the graphical elements inside or on the boundary of the protection region are drawn. When a protection region is set for shielding, only the portions of the graphical elements outside the protection region and its boundary are drawn.

Several clip and shield regions may be in effect simultaneously. In this case, only portions of the graphical elements inside or on the boundary of the intersection of all individual clip regions and outside the union of

all individual shield regions are drawn. See figure 3.

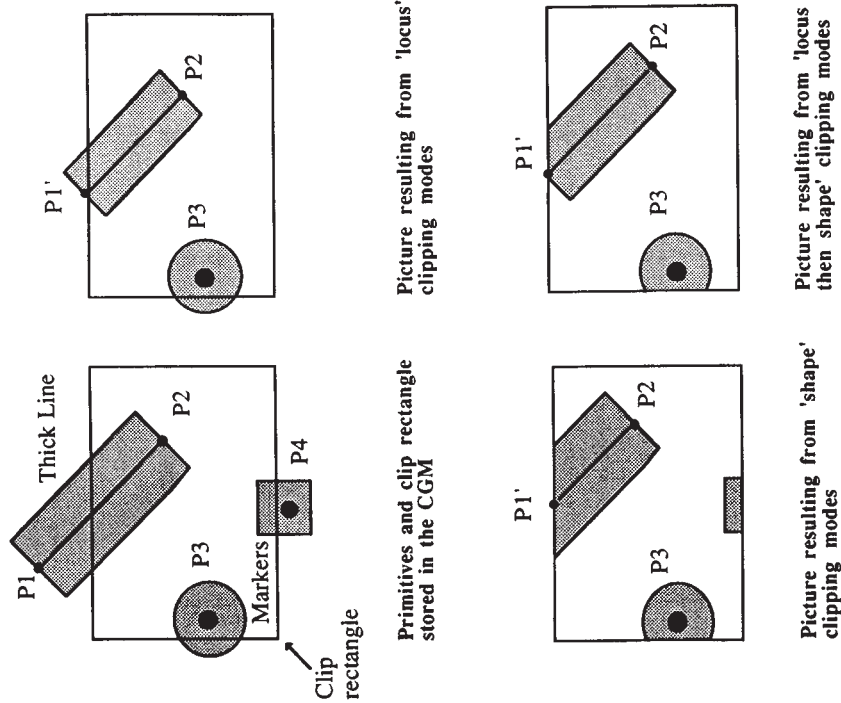


Figure 2 — Clipping modes

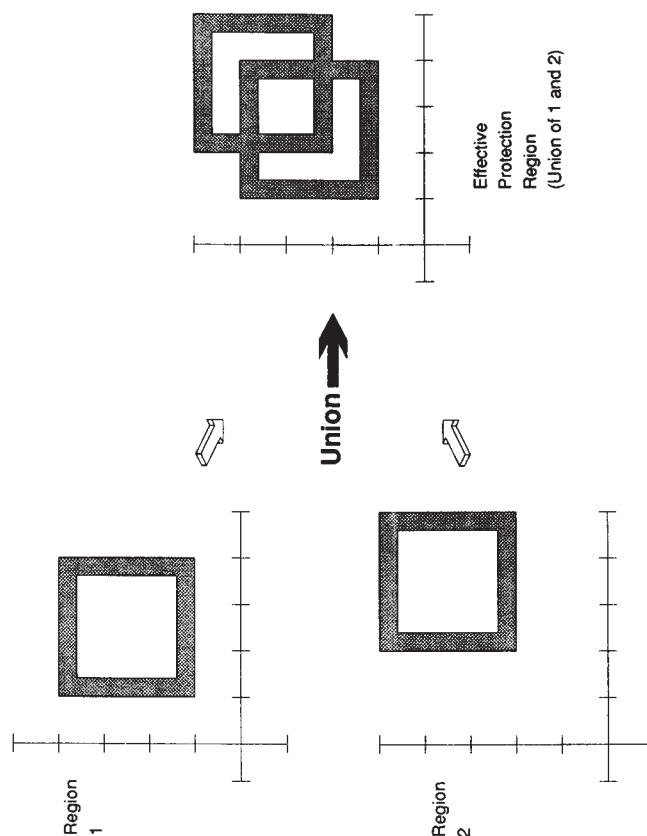


Figure 3 — Combining protection regions (hatching denotes the region)

4.5.5 Generalized text path

The GENERALIZED TEXT PATH MODE element selects the method for placing the text along the text path. When the mode is 'off', the text is displayed along the text path defined by the CHARACTER ORIENTATION and TEXT PATH elements.

When GENERALIZED TEXT PATH MODE is 'non-tangential', the characters are drawn along the last compound text path (defined by the BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH elements), but the character orientation vectors are not rotated relative to the text path. When GENERALIZED TEXT PATH MODE is 'axis-tangential', the characters are positioned along the last compound text path (defined by the BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH elements) and the character orientation vectors for each character are rotated by an amount equal to the angle of the tangent to the text path at the character position.

Illustrations of GENERALIZED TEXT PATH MODE are shown in figure 21 through figure 26.

4.5.6 Mitre limit

The control elements include an element for controlling how the joins of lines and filled-area edges are drawn. Line or edge joins may be rendered with mitre join style, as determined by the appropriate line and edge attribute elements. The mitre join is formed by projecting the outer edges of the lines or edges at the corner until the projections meet at a point. When mitre joins are being rendered, there is the possibility of the mitre projecting very far if the line segments meet at a very sharp angle at the vertex. The MITRE LIMIT provides a means of specifying that long mitres are to be truncated at some point to form a flat bevel. *Mitre length* is defined to be the distance from the point at which the inside edges of adjoining line segments meet to the point at which the outside edges meet. The parameter of MITRE LIMIT is a single real number. If the mitre length divided by the line width or edge width exceeds the value of the parameter of the MITRE LIMIT element, then the mitre join is truncated at that limit. However, if the truncation would result in a flat bevel within the triangular notch of the corresponding bevel join style, then the bevel join style is used.

4.5.7 Transparent cell colour

The control elements include an element, TRANSPARENT CELL COLOUR, which allows certain parts of CELL ARRAY elements, TILE and BITONAL TILE elements, and the filling patterns defined by PATTERN TABLE elements to be designated as transparent.

The usual way of rendering these elements is to draw each cell with the colour designated by its associated cell colour specifier. TRANSPARENT CELL COLOUR allows a given value of a cell colour specifier (indexed or direct) to be defined to be transparent. Any cell whose cell colour specifier matches this value is not drawn when the primitive is rendered.

In order to invoke the transparency effect in an element, the cell colour specifier shall have the transparent designation at the time of occurrence of the affected element in the metafile. If a cell in a pattern definition is to be designated as transparent, then the colour value of the associated cell colour specifier shall be defined as transparent prior to the occurrence in the metafile of the PATTERN TABLE element which defines the particular pattern. That is, the TRANSPARENT CELL COLOUR element only affects elements which occur after it in the metafile and has no effect on previously occurring elements or previously drawn primitives.

The rendering of any element affected by TRANSPARENT CELL COLOUR shall not involve more than one distinct value of cell colour specifier simultaneously being designated as transparent.

NOTE — It is recommended that metafile generators limit themselves to using a single value of the cell colour specifier as their transparent cell value.

Graphical primitive elements

Concepts

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 CGM provides the graphical primitive elements

POLYLINE
DISJOINT POLYLINE
POLYMARKER
TEXT
RESTRICTED TEXT
APPEND TEXT
POLYGON
POLYGON SET
CELL ARRAY
GENERALIZED DRAWING PRIMITIVE (GDP)
RECTANGLE
CIRCLE
CIRCULAR ARC 3 POINT
CIRCULAR ARC 3 POINT CLOSE
CIRCULAR ARC CENTRE

CIRCULAR ARC CENTRE CLOSE
ELLIPSE
ELLIPTICAL ARC
ELLIPTICAL ARC CLOSE
CIRCULAR ARC CENTRE REVERSED
CONNECTING EDGE
HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER
POLYSYMBOL
BITONAL TILE
TITLE

The metafile supports access to special geometric output capabilities of devices and workstations through the GDP. The GDP has a list of points in VDC as a parameter. It is thus well suited for non-standardized output primitives, which have position, shape, extent, etc., whereas ESCAPE is better suited for non-standardized device control functions.

The formal definition of the CGM describes graphical primitive elements which are positionally independent by virtue of containing complete explicit positional information within each element definition.

The TEXT, RESTRICTED TEXT, and APPEND TEXT elements and related text attribute elements are defined in the current VDC space. Thus, they are affected by changes to the Virtual Device Coordinate format.

The following types or categories of graphical primitive elements are defined for the CGM: line elements, marker element, text elements, filled-area elements, cell elements, and symbol elements.

The line elements are

POLYLINE
DISJOINT POLYLINE
CIRCULAR ARC 3 POINT
CIRCULAR ARC CENTRE
ELLIPTICAL ARC
CIRCULAR ARC CENTRE REVERSED

CONNECTING EDGE
HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER

The marker element is

POLYMARKER

Graphical primitive elements

Concepts

The text elements are

TEXT
RESTRICTED TEXT
APPEND TEXT

The filled-area elements are

POLYGON
POLYGON SET
RECTANGLE
CIRCLE
ELLIPTICAL ARC CLOSE
CIRCULAR ARC 3 POINT CLOSE
CIRCULAR ARC CENTRE CLOSE
ELLIPSE

The cell elements consist of one cell array element and two tile array elements:

CELL ARRAY
BITONAL TILE
TITLE

The symbol element is

POLYSYMBOL

In addition to these classes of elements, the GENERALIZED DRAWING PRIMITIVE (GDP) is a graphical primitive element that may be used to access device (or implementation) specific graphical primitives that are not accessed by the standardized elements. In addition to the graphical primitive elements listed above, this part of ISO/IEC 8632 defines elements permitting the definition of 'compound primitives' from instances of one or several of the other graphical primitives. The following classes of compound primitives are defined: 'compound text', 'closed figure' and 'compound line'. The elements that may be used to specify compound primitives are listed in table 1.

Table 1 — Contributing primitives to compound primitives

Compound Primitive	First Element	Primitives Included	Other Elements	Final Element
Compound Text	TEXT, RESTRICTED TEXT (Note 1)	APPEND TEXT (Note 2)		APPEND TEXT (Note 3)
Closed Figure	BEGIN FIGURE	Line Primitives, Fill Primitives (Note 4), GDP (Note 5)	NEW REGION	END FIGURE
Compound Line	BEGIN COMPOUND LINE	Line primitives, GDP (Note 5)		END COMPOUND LINE
Tile Array	BEGIN TILE ARRAY	BITONAL TILE, TILE		END TILE ARRAY

NOTES

- 1 The final/not final flag is 'not final'; the primitive defines the reference point of the entire compound text primitive; the text of the primitive is accumulated.

ISO/IEC 8632-1:1992 (E)	ISO/IEC 8632-1:1992 (E)	Graphical primitive elements	Graphical primitive elements	Concepts
Concepts	2 The final/hot final flag is 'not final'. 3 The final/hot final flag is 'final'; the text of the primitive is accumulated before the compound primitive is closed. 4 All primitives of the identified classes may be included. 5 Whether and how a GDP may contribute to closed figure or compound line, and whether or how it specifies that the figure open state or compound path state be opened, maintained or closed, is specified with the definition of the GDP in the International Register of Graphical Items. With the exception of tile array, graphical primitives and compound primitive elements may be subject to transformation in VDC space (segment and copy transformation, see 4.10.4.2 and 4.10.5). Such a transformation may change the shape of some primitives. If there is a skew, a primitive initially specified as a rectangle may become a parallelogram. If there is an anisotropic scaling, a primitive initially specified as a circle may become an ellipse. Note that the shape of markers is not affected by such transformations. Anisotropic transformation will change the angle at which non-parallel lines intersect: isotropic transformation will preserve the angle at which non-parallel lines intersect.	POLYBEZIER	4.6.1.2 Compound line	parameterization is described in 5.6.25, and the principle underlying the definition of the element are described in 4.6.10.1. generates a sequence of one or more cubic Bezier curves; the parameterization is described in 5.6.26 and the principles underlying the definition of the element are described in 4.6.10.2.
4.6.1 Line elements	There are two general line elements — POLYLINE and DISJOINT POLYLINE — as well as curve elements that define conic arcs (circular, elliptical, parabolic, and hyperbolic arcs) and elements that define spline curves (B-splines and Beziers). There is also a compound line element, whose definition is composed of these individual line elements.	4.6.1.1 Description	4.6.1.3 Attributes	The appearance of all line elements is controlled by the line attributes, the LINE BUNDLE INDEX, and those ASPECT SOURCE FLAGS which are associated with the line attributes that may be bundled. These are described in 4.7.1.
POLYLINE	generates a set of connected lines as defined by a list of points, starting with the first, drawing a line through each successive point, ending at the last point.	DISJOINT POLYLINE	4.6.1.4 Usage of line elements	POLYLINE is the most general of the primitives. DISJOINT POLYLINE is intended for situations where the alternative would be a large number of 2-point POLYLINE elements. The conic arc primitives (circular, elliptical, hyperbolic, and parabolic) and spline primitives (Polybezier, Non-uniform B-splines, and NURBS) provide data compression by comparison with POLYLINE and allow the arcs to be described without knowledge of the resolution of the final viewing surface.
CIRCULAR ARC xxx	generates a single circular arc; two parameterizations of the arc are possible; these are described in 5.6.13 and 5.6.15. A reverse direction arc can also be specified; see 5.6.20.	ELLIPTICAL ARC	4.6.1.5 Clipping of line elements	In Version 2 and Version 3 metafiles, line clipping is controlled by the LINE CLIPPING MODE element, which can have one of the following values: 'locus', 'shape', or 'locus then shape'. However, clipping applies only if the CLIP INDICATOR is 'on'.
CONNECTING EDGE	generates a single elliptical arc; the parameterization of the arc is described in 5.6.18. a line segment connecting the last point of the preceding line element to the next point is generated during the construction of a closed figure. The next point is either the first point of the next line element or the current closure point.	HYPERBOLIC ARC	For 'locus' clipping, the mathematical locus of the line is clipped at the intersection with the clip rectangle before shape rendering is applied. Hence, part of the shape of a clipped line may appear outside the clip rectangle.	For 'shape' clipping, the shape of the rendered line is clipped to the intersection with the clip rectangle; that is, nothing is drawn outside the clip rectangle. A portion of a widened line may appear inside the clip rectangle even though the mathematical locus of the line itself may be entirely outside the clip rectangle.
PARABOLIC ARC	generates a hyperbolic arc; the parameterization is described in 5.6.22, and the principles underlying the transformable parameterization are described in 4.6.8. generates a parabolic arc; the parameterization is described in 5.6.23, and the principles underlying the transformable parameterization are described in 4.6.9.	NON-UNIFORM B-SPLINE	4.6.1.6 Transformation of line elements	If the LINE WIDTH SPECIFICATION MODE has the value 'absolute', then all line aspects — line width, line cap, line join, line dash and gap lengths — are subject to the VDC-to-Device mapping (see 4.4.7) as well as to both segment and copy transformation (see 4.10.4.2 and 4.10.5). Note that the entire locus of an arc is subject to these transformations. In the case of an anisotropic mapping or transformation, the
NON-UNIFORM RATIONAL B-SPLINE	generates a Non-Uniform Rational B-Spline (NURBS) curve; the	33	34	

Graphical primitive elements

Concepts

rendered width of the line will change with the direction of the line segment.

If the line width is specified in mode 'scaled', 'fractional', or 'mm', it is not affected by any transformations.

4.6.2 Marker element

4.6.2.1 Description

There is a single marker element.

POLYMARKER

generates markers of a specific type at each of a list of points.

4.6.2.2 Attributes

The appearance of the markers is controlled by the marker attributes, the MARKER BUNDLE INDEX and the ASPECT SOURCE FLAGS associated with the marker attributes that may be bundled. These are described in 4.7.2.

4.6.2.3 Clipping of the marker element

The following discussion applies to Version 1 metafiles. Markers conceptually indicate the location of their specifying points. Therefore, if the value of CLIP INDICATOR is 'on', the marker is visible if, and only if, its specifying point is within the rectangle specified by CLIP RECTANGLE. If its specifying point is inside the clip rectangle but part of the marker lies outside, the manner in which the marker is clipped (or not clipped) is not standardized for Version 1 metafiles.

NOTE — When it is required, in Version 1 metafiles, to make visible those parts of a marker that are inside the clip rectangle, when the position of the marker is outside the rectangle, then the most appropriate primitive element is the TEXT element, used with a single-character string argument. Three PRECISIONs are available to control the precision with which TEXT elements are clipped. STROKE precision requires clipping even within the body of a text character. Centring of the text character at the specifying position may be achieved with the TEXT ALIGNMENT element.

In Version 2 and Version 3 metafiles, marker clipping is controlled by the MARKER CLIPPING MODE element, which can have one of the following values: 'locus', 'shape' or 'locus then shape'. However, clipping applies only if the value of CLIP INDICATOR is 'on'.

For 'locus' clipping, the specifying points of each marker are clipped at the intersection with the clip rectangle before shape rendering is applied. The marker is only visible if its specifying point is within the clip rectangle. Hence, part of the shape of a marker may appear outside the clip rectangle providing its specifying point is within the clip rectangle.

For 'shape' clipping, the shape of the rendered marker symbols are clipped to the intersection with the clip rectangle; that is, nothing is drawn outside the clip rectangle. Portions of the marker symbol may appear inside the clip rectangle even if the marker's position is outside.

For 'locus then shape' clipping, the clipping is first applied to the specifying points of each marker, as with 'locus' clipping, and then subsequently the rendered shape of the markers are again clipped.

4.6.2.4 Transformation of the marker element

If the MARKER SIZE SPECIFICATION MODE has the value 'absolute', then marker size is subject to the VDC-to-device mapping (see 4.4.7) as well as to both segment and copy transformation (see 4.10.4.2 and 4.10.5). The shape of markers is never affected by transformations; for example, a circle used as a marker type shall always appear as a circle. Only the marker size may be transformed. Conceptually, vectors with length equal to the marker size but at all possible orientations are transformed; the marker size is the maximum length of the vectors under the transformation.

Graphical primitive elements

Concepts

If the marker size is specified in mode 'scaled', 'fractional', or 'mm' it is not affected by any transformations.

4.6.3 Text elements

4.6.3.1 Description

Three text elements are provided.

TEXT

generates a text string (or part of a text string) aligned to a particular point.

RESTRICTED TEXT

generates a text string (or part of a text string) that is constrained within a given area.

APPEND TEXT

generates a part of a text string which was started with a TEXT or RESTRICTED TEXT element.

4.6.3.2 Attributes

The appearance of text elements is controlled by the text attributes, the TEXT BUNDLE INDEX, and those ASPECT SOURCE FLAGS which are associated with the text attributes that may be bundled. These are described in 4.7.3.

Changes to certain of the text attributes, as listed in the description of APPEND TEXT (see 5.6.6), are permitted between a non-final text element and its succeeding APPEND TEXT element.

4.6.3.3 Usage of text elements

Each text element has a 'final/not-final' flag. This permits a text string to be started with a TEXT or RESTRICTED TEXT element and continued with one or more APPEND TEXT elements. The last element will have its flag set to 'final'. The initial element is always TEXT or RESTRICTED TEXT; subsequent elements may only be APPEND TEXT.

The current setting of TEXT ALIGNMENT is used to align the complete text string assembled from the separate text elements.

4.6.3.4 Clipping of text elements

Clipping of text strings is described in 4.7.3.2.

4.6.3.5 Transformation of text elements

The vectors specified by the CHARACTER ORIENTATION element (see 4.7.3.2) are subject to the VDC-to-Device mapping (see 4.4.7) as well as to both segment and copy transformation (see 4.10.4.2 and 4.10.5).

4.6.4 Filled-area elements

4.6.4.1 Description

There are two general fill elements: POLYGON and POLYGON SET. In addition, there are several filled-area elements that correspond to the basic geometric shapes — circles, rectangles, pie sectors, etc. These metafile representations of the common geometric entities are compact, scalable, and independent of the resolution of the final viewing surface. Finally, there is also a compound filled-area element — Closed Figure — whose definition is composed of a sequence of individual line and filled-area primitives (see 4.6.11).

POLYGON

generates an area and its edge, defined by a list of points; the style of the area is one of 'hollow', 'solid', 'pattern', 'hatch', 'empty', 'geometric pattern', or 'interpolated'; the visibility and style of the

Concepts

Graphical primitive elements

edges of the area depend on the edge attributes alone.

POLYGON SET

generates a number of areas and their edges, defined by a list of vertex points and vertex flags; the set of styles is the same as for POLYGON; the vertex flags indicate the different polygons in the set; the vertex flags and the edge attributes together control the visibility and style of individual edge segments of each polygon.

RECTANGLE

generates an upright rectangular area; the set of styles is the same as for POLYGON.

CIRCLE

generates a circle; the set of styles is the same as for POLYGON.

CIRCULAR ARC xxx CLOSE

generates a partial circular area; 'pie' and 'chord' style arcs are possible; two parameterization of the arcs are provided; these are described in 5.6.14 and 5.6.16; the set of styles is the same as for POLYGON.

ELLIPSE

generates an ellipse; the parameterization of the ellipse is described in 5.6.17; the set of styles is the same as for POLYGON.

ELLIPTICAL ARC CLOSE

generates a partial elliptical area; 'pie' and 'chord' style arcs are possible; the parameterization is described in 5.6.19; the set of styles is the same as for POLYGON.

4.6.4.2 Attributes

The appearance of all filled-area elements is controlled by the fill attributes, the FILL BUNDLE INDEX, the EDGE BUNDLE INDEX, and the ASPECT SOURCE FLAGS associated with the fill attributes that may be bundled. These are described in 4.7.4.

4.6.4.3 Usage of fill elements

POLYGON provides for the representation of standard irregular areas. RECTANGLE, because it is upright, is a more efficient parameterization of a rectangle than a POLYGON and may be implemented directly in some systems.

The circular and elliptical fill primitives, as well as closed figure fill primitives incorporating such line primitives as the conic arc elements and spline curve elements (see 4.6.11), provide an efficient parameterization and allow the areas to be produced accurately without knowledge of the resolution of the final viewing surface.

POLYGON SET allows a related set of polygons to be represented. All attributes of each of the polygons are the same. The specification of the vertex flags allows disjoint polygons (such as both the body and the dot of the letter 'i'), holes (as in a broad ring) and overlapping areas. Accurate rendering of abutting areas of uniform or graded colour, pattern or hatch, and control over individual edge segment visibility, are possible.

4.6.4.4 Interior

The interior of a filled-area element is defined as follows. For a given point, create a straight line starting at that point and going to infinity. If the number of intersections between the straight line and the filled area boundary is odd, the point is within the filled area; otherwise it is outside. If the straight line passes a filled-area vertex tangentially, the intersection count is not affected. If a point is within the filled area, it is included in the area to be filled subject to the rules for boundaries and edges (see 4.7.4.3).

4.6.4.5 Edges

The edge of a filled-area element can be either visible or invisible. If visible, the individual edge attributes or the EDGE BUNDLE INDEX (according to the edge ASF values) govern the appearance.

Graphical primitive elements

Concepts

If the edge is visible, it is drawn on top of the interior — the edge has precedence over the interior when drawn and will always be fully visible. The boundary drawn for style 'hollow' is considered as the representation of the interior. While the edge has precedence, the boundary may be partly visible as well. Parts of edges which are clipped become invisible — clipping of edges is identical to clipping of line elements. Parts of interiors which are clipped will, in the case of style 'hollow', have a boundary drawn at the clipping boundary.

The "realized edge" is defined to be the zero-width ideal boundary line of the filled-area if the edge is invisible, and the finite-width displayed line if the edge is visible. ISO/IEC 8632 does not mandate the alignment of the finite-width realized edge with respect to the zero-width ideal edge (i.e., whether the former is centred on the latter or aligned some other way such as inside).

The "realized interior" is defined as extending to and terminating at the realized edge. The discussion of interior in the remainder ISO/IEC 8632 should be considered to pertain to realized interior.

4.6.4.6 Clipping

If parts of a filled-area element are clipped, then the intersection of the interior and the clip boundary becomes part of the boundary of the resulting clipped area for the purposes of display of the boundary for interior style 'hollow'. If the edge is visible, it is not drawn along the new boundary segments created by the clipping of the area. Edge clipping is controlled by the EDGE CLIPPING MODE element, which has the same enumerations as LINE CLIPPING MODE. Edges are clipped in the same way that lines are clipped; see 4.6.1.6.

4.6.4.7 Transformation

The entire mathematical locus of rectangles, circular and elliptical filled-area elements is subject to the VDC-to-Device mapping (see 4.4.7), segment transformations (see 4.10.4.2) and copy transformations (see 4.10.5). Because anisotropic transformation does not preserve angles between non-parallel lines, rectangles may become parallelograms and circles may become ellipses.

If the INTERIOR STYLE SPECIFICATION MODE is 'absolute', the geometric aspects of fill interiors are subject to all transformations. These aspects include PATTERN SIZE; direction, spacing, and width of hatch lines; and reference geometry of interpolated interior. If the mode is 'scaled', 'fractional', or 'mm', then none of these aspects is subject to transformation.

Geometric aspects of edges are treated in exactly the same way as the corresponding aspects of lines.

4.6.5 Cell elements

The cell elements comprise a single Cell Array element and two Tile Array elements.

4.6.5.1 Cell array element

The single cell array element is:

CELL ARRAY

represents a 2-dimensional array of colour values, which cover a rectangle or parallelogram.

The colour values are either direct colour values or indexes into the COLOUR TABLE, according to the current COLOUR SELECTION MODE. The colour values are in the precision declared by a *local colour precision* parameter of the CELL ARRAY element.

CELL ARRAY has no associated attributes.

4.6.5.2 Tile Array Elements

A Tile Array is a compound raster image primitive, whose definition is delimited by the BEGIN TILE ARRAY and END TILE ARRAY delimiter elements. Between the delimiter elements is a series of equally

Concepts

sized individual "tiles" which all together form a contiguous rectangular block of tiles. Each tile is defined by a non-overlapping TILE or BITONAL TILE element. The first tile is placed at the *position* parameter of the BEGIN TILE ARRAY element. Any subsequent tiles are placed in order first in the cell path direction and then in the line progression direction. The tile positions are numbered as shown in figure 4.

The elements defining the tiles which comprise a Tile Array are

BITONAL TILE: defines a rectangular raster image, either uncompressed or compressed according to a selected compression method. Only two colours are used to define the image. Each cell is associated with one of the colour indexes 0 or 1, and the colour values associated with 0 and 1 are defined locally by each BITONAL TILE element.

TILE: defines a rectangular raster image, either uncompressed or compressed according to a selected compression methods. The colours associated with the cells may either be bitonal or full colour, may be specified by either indexed or direct mode, and are specified according to the applicable colour precisions and modes.

The TILE element contains a *cell colour precision* parameter, which behaves as does the local colour precision of CELL ARRAY (see previous section). BITONAL TILE does not contain this parameter, as its colours are always represented at 1-bit precision prior to compression.

The TILE and BITONAL TILE elements are not independent graphical primitives as are POLYLINE and CELL ARRAY. They contain no positioning or dimensioning information. Each element contains only the raster content of a single raster tile and any control parameters which apply to that tile. The complete Tile Array primitive is formed by one or more tiles between BEGIN TILE ARRAY and END TILE ARRAY. BEGIN TILE ARRAY contains all parameters which apply to the collection of tiles (if there is more than one tile). The parameters apply uniformly to each tile in the collection.

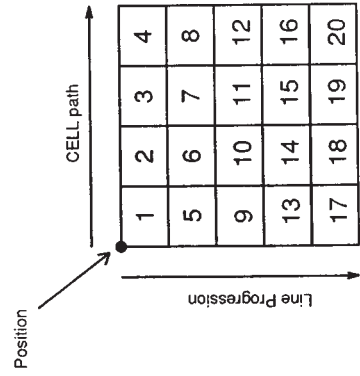


Figure 4 — Ordering and layout of tiles

4.6.5.2.1 Relationship to CELL ARRAY. Both tile arrays and cell arrays are composed of cells. While cells in cell arrays are subject to all transformations, cells in tiles are always axis aligned and rectangular. They scale to the display surface but do not otherwise transform. As with CELL ARRAY, the position point of a tile array corresponds with the corner of the first cell (rather than the centre of the cell).

4.6.5.2.2 Allowable states for tiles and tile array elements. The tile elements, TILE and BITONAL TILE, may appear only in Tile Array State (TAS). Tile Array compound elements, delimited by BEGIN TILE ARRAY and END TILE ARRAY, may appear only in Picture Open State. They may not appear in segments or other compound primitive definitions.

Concepts

4.6.5.2.3 Compressed cell data. The cell colour data of the tile elements is a compressed stream of cell colour specifiers. The datatype is Bitstream. For the BITONAL TILE element the Bitstream parameter consists of a sequence of 1-bit binary colour indexes which are compressed by the selected technique (the list of techniques includes 'bitmap' which is uncompressed). The resulting compressed binary data object is the parameter of the element. Each of the CGM encodings (Binary, Character, and Clear Text) define a technique for representing and encoding the compressed binary data object.

4.6.5.2.4 Tiling. The tiling mechanism specified is based on the Tiled Raster Interchange Format that has been developed for ISO/IEC 8613-7. Definition of a tile array is initiated by the BEGIN TILE ARRAY delimiter element and terminated by the END TILE ARRAY element. During tile array definition subsequent tile elements define individual tiles within the tiled image. The number of tiles is determined by the parameters of the BEGIN TILE ARRAY element. A tile array contains one or more tiles.

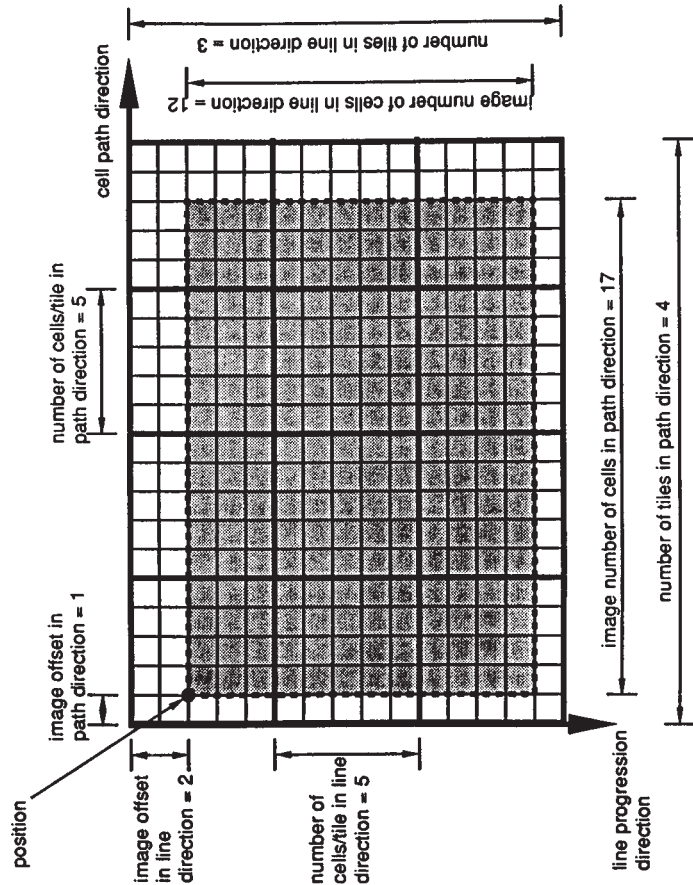


Figure 5 — Relationship of the image to the tile space

The number of tiles defined during tile array definition shall match the number indicated by the BEGIN TILE ARRAY element. Annex D contains recommendations for interpreter fallback in the case that the tiles are missing.

Concepts

Graphical primitive elements

The tiling offset and size parameters define the position of the actual image within the tile space, relative to the *position* parameter of the BEGIN TILE ARRAY element. This is the portion of the whole tile array which contains information. There may be a border of unused cells surrounding the actual image. These cells contain no information and should not be drawn by interpreters. These unused cells are however included in the encoding of those tiles which overlap the border and the defined image. See figure 5 for the relationship of the actual image and the total tile space.

NOTE — The unused cells must have some value assigned, and it is recommended that they all be set identically to the background colour.

4.6.6 Circular arc elements

The CGM provides for two forms of specification of circular arc elements: a centre-radius specification and a 3-point specification. Each has its advantages and disadvantages with respect to numerical accuracy, relationship of defining data to the VDC range, etc.

NOTE — When choosing which parameterization to use, one should decide where possible numerical inaccuracy would be least disturbing. The 3-point form specifies exact arc endpoints, but might result in inaccurate centre-point calculations, whereas the centre form specifies exact centre point but might result in roundoff errors on the ends of the arc. The 3-point form would thus be more appropriate for smoothly joining an arc to a polyline in a line drawing, whereas the centre form would be more appropriate for pie charts.

4.6.7 Elliptical elements

4.6.7.1 Geometric concepts

Ellipses are specified by Conjugate Diameter Pairs. A Conjugate Diameter Pair (CDP) of an ellipse is a pair D_1 of diameters of the ellipse such that a tangent to the ellipse at each endpoint is parallel to the other diameter. The four tangents to the ellipse at the endpoint of the CDP thus form a parallelogram whose sides are bisected by the endpoints of the diameters.

Any CDP of the ellipse remains a CDP across any graphical transformation which transforms an ellipse into an ellipse. This is demonstrated in figure 6 in which the ellipse has been scaled by a factor of two in the y -direction only.

Thus any CDP of a desired ellipse can be used to specify the ellipse. Note that the (mutually perpendicular) major and minor axes of an ellipse and any pair of perpendicular diameters of a circle are CDP's, although they do not necessarily remain perpendicular across a transformation.

Thus to specify an ellipse, all that is needed is three points:

- the centrepoint of the ellipse;
- two CDP endpoints (one endpoint from each diameter).

4.6.7.2 Parameterization of elliptical elements in CGM

The ellipse itself in each of the three elliptical elements is parameterized as in the preceding section, the centrepoint and two CDP endpoints. For the two elliptical arc elements, the start and end of the defined arc section is parameterized by two semi-infinite rays originating at the centrepoint. The intersection of these rays with the ellipse defines two points on the ellipse, and these two points define the arc. The conjugate diameters parameterization of ellipses and elliptical arcs has the property of being transformable — the ellipse defined by the transformed parameter data is the transformed ellipse. The conjugate diameter parameterization has other useful properties as well.

For simplicity, consider the ellipse that is centred at the origin, and let P_1 and P_2 designate the endpoints of the conjugate diameters. Let M be the 2×2 matrix whose first column is P_1 and whose second column is P_2 . The transformation M maps points on the unit circle centred at the origin ($x^2 + y^2 = 1$) onto the ellipse.

41

Graphical primitive elements

Concepts

The unit circle is referred to as the "canonical ellipse". If the ellipse is non-degenerate, then M is non-singular, hence invertible, and M^{-1} maps points on the ellipse onto points on the unit circle centred at the origin. M maps the unit vectors u_1 and u_2 respectively onto P_1 and P_2 , where u_1 and u_2 are vectors from the origin to the points (1,0) and (0,1) respectively. These principles generalize easily to ellipses which are not centred at the origin — there is a translation term in the mapping so that the transformation is not linear but is affine.

4.6.8 Hyperbolic Arc Element

The CGM parameterization of the hyperbolic arc closely parallels that of the ellipse (see figure 7). The "canonical hyperbola" is defined by $x^2 - y^2 = 1$. It passes through the point (1,0). At (1,0) the tangent to the hyperbola is parallel to the vector u_2 , which is the vector from the origin to the point (0,1). The canonical hyperbola has "centre" (the point where the asymptotes cross) at the origin. For any non-degenerate hyperbola centred at the origin, there is a linear transformation which maps the canonical hyperbola onto the given hyperbola. This transformation maps the points (1,0) and (0,1) respectively onto a pair of points P_1 and P_2 . In this case, P_1 is on the hyperbola but P_2 is not. At P_1 the tangent to the hyperbola is parallel to the line from the origin to P_2 . The asymptotes of the hyperbola are parallel to the vectors $P_1 + P_2$ and $P_1 - P_2$. Points with such properties are referred to as a the *conjugate radius endpoint* and *transverse radius endpoint* respectively. The *transverse radius endpoint* is the one which lies on the hyperbola; the *conjugate radius endpoint* does not. These points (plus the centre point) parameterize the hyperbola in CGM.

As with the ellipse, if the matrix M is formed whose columns are the points P_1 and P_2 , then this is the invertible transformation which maps points on the canonical hyperbola onto points on the given hyperbola (and whose inverse maps the given hyperbola onto the canonical hyperbola). The generalization to hyperbolas whose centre is not the origin is straight forward.

As with elliptical arcs, the start and end of the hyperbolic arc are parameterized by vectors from the centre.

In both the case of the ellipse and the case of the hyperbola, the conjugate parameterizations can be derived from x - y implicit equations and vice-versa.

4.6.9 Parabolic arc element

The principles used to parameterize elliptical arcs are also used to parameterize parabolic arcs, but the analogy is not quite as strong between parabolic arc and elliptical arc as it is between hyperbolic arc and elliptical arc. The parameterization is again in terms of a transformation of a "canonical parabola". In this case, the canonical parabola is $2(x+y) = (x-y)^2 + 1$ for $x \leq 1$ and $y \leq 1$. This parabolic arc is symmetric about the line $y=x$, has endpoints (1,0) and (0,1), is tangent to the x -axis and y -axis respectively at these points, and remains entirely in the first quadrant. See figure 8.

The general parabolic arc is parameterized by the endpoints of the arc, P_1 and P_2 and the intersection of the tangents to the arc at the endpoints. This intersection point is called the "centre" of the parabolic arc, C . Define $V_1 = P_1 - C$ and $V_2 = P_2 - C$, and form the 2×3 matrix M whose first column consists of the components of V_1 , second column consists of the components of V_2 , and third column consists of the components of C . For non-degenerate parabolic arcs M is an affine transformation that maps points on the canonical parabolic arc onto points on the given parameterized parabolic arc.

42

Concepts

ISO/IEC 8632-1:1992 (E)
Graphical primitive elements

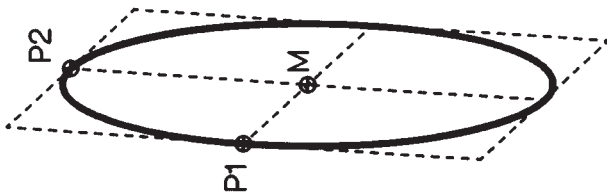
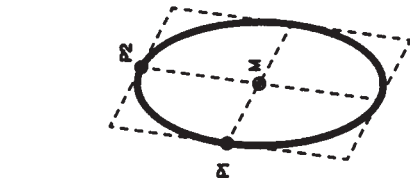
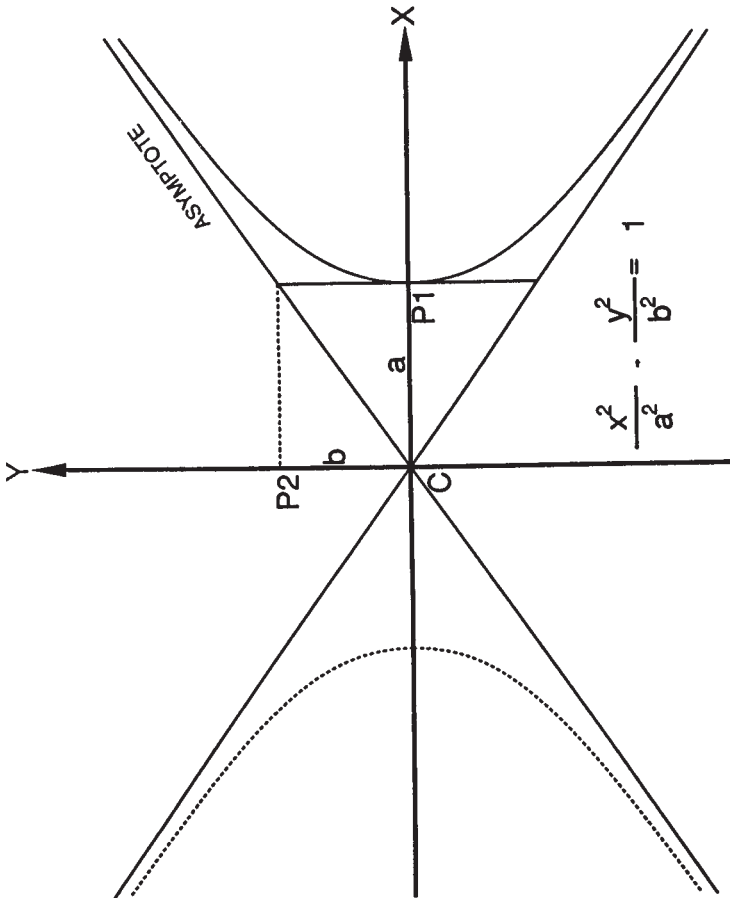


Figure 6 — Anisotropic scaling of an ellipse

Concepts

ISO/IEC 8632-1:1992 (E)
Graphical primitive elements



C - centre
P1 - transverse radius point
P2 - conjugate radius point

Figure 7 — Example of hyperbolic arc

4.6.10 Spline curve elements

The CGM provides three spline curve elements: non-uniform B-splines; non-uniform rational B-splines; and Bezier curves.

4.6.10.1 Non-uniform B-splines

The CGM provides both rational and non-rational B-splines of varying orders.

4.6.10.1.1 Parameterization. The non-uniform B-spline is parameterized by a spline order, a list of knots, a list of control points, and parameter range limits defining the curve section to be drawn.

The non-uniform rational B-spline is parameterized by a spline order, a list of knots, a list of control points, a list of weights associated with the control points, and parameter range limits defining the curve section to be drawn.

4.6.10.1.2 Mathematical definition. The non-uniform B-spline is expressed parametrically in the form

Equation 1 — Non-uniform B-spline

$$G(t) = \sum_{i=0}^n P_i B_i^k(t)$$

where

n — number of control points;

P_i — control points, (P_x, P_y) ;

B_i^k — B-spline basis functions defined by order k and knot vector T .

The knot vector consists of a non-decreasing sequence of real numbers $(T_1, \dots, T_{n+k})$ such that $T_i \leq T_{i+1}$ for all $i = 1, \dots, n$.

The curve itself is defined for the range $T_k \leq t < T_n$ and can be confined to the range $[T_{\min}, T_{\max}]$, where $T_k \leq T_{\min} \leq T_{\max} < T_n$. $T_{\min}$ and $T_{\max}$ are specified as part of the non-uniform B-spline primitive.

The B-spline basis functions are defined by the recursive relation:

$$B_i^1 = \begin{cases} 1 & \text{if } T_i \leq t \leq T_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_i^k(t) = \begin{cases} 0, & \text{if } t < T_i \text{ or } t \geq T_{i+k} \\ \frac{(t - T_{i-k}) * B_{i-1}^{k-1}(t)}{T_i - T_{i-k}} + \frac{(T_{i+1} - t) * B_{i+1}^{k-1}(t)}{T_{i+1} - T_{i+k}}, & \text{if } T_i \leq t < T_{i+k} \end{cases}$$

For multiple identical knot values, the fractional terms above may evaluate to $\frac{0}{0}$. In such cases the terms are defined to be 0.

A non-rational non-uniform B-spline curve is generalized by using equation 1 with a set of control points

$$P_i = (P_x, P_y, P_w) \text{ for } i = 1, \dots, n.$$

A rational non-uniform B-spline is expressed parametrically in the following form:

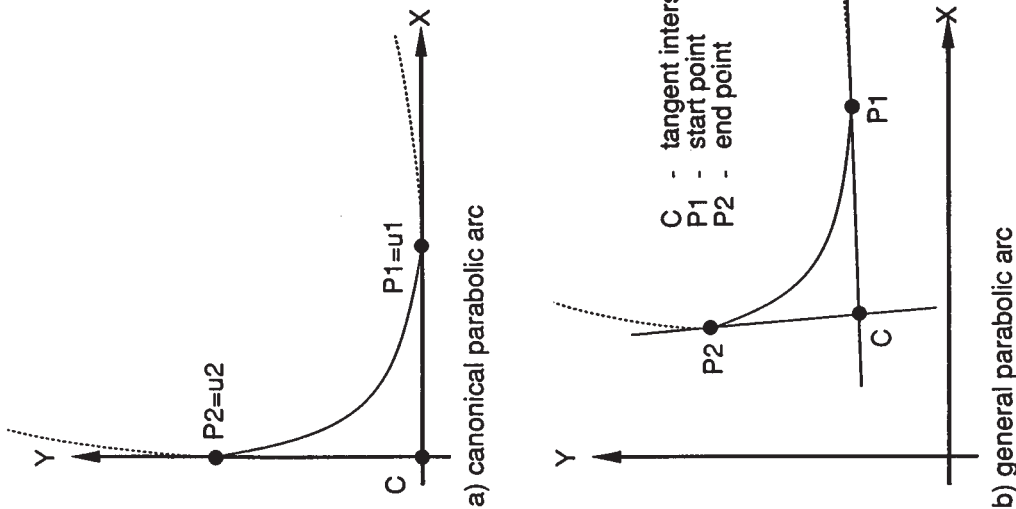


Figure 8 — Example of parabolic arc

Equation 2 — Rational non-uniform B-spline

$$G(t) = \frac{\sum_{i=0}^n B_i^k(t) P_i w_i}{\sum_{i=0}^n B_i^k(t) w_i}$$

Given a set of control points $\{P_i\}$ and weights $\{w_i\}$, this is equivalent to creating a set of homogeneous control points $\{w_i P_i, w_i\}$ in 3D, using equation 1 in 3D and projecting the resulting points back into 2D, projecting the homogeneous point onto the $\{0,0,1\}$ plane in 3D. This is achieved by dividing the first two (homogeneous) coordinates by the third.

4.6.10.2 Polybezier

This element defines one or more cubic Bezier curves.

4.6.10.2.1 Parameterization. The polybezier is parameterized by a list of points and an indicator specifying the degree of continuity between the individual Bezier curves. If there are $n(n \geq 1)$ Bezier curves, then the data list will contain:

- $4n$ points if the continuity parameter is *discontinuous*;
 - $3n+1$ points if the continuity parameter is *continuous*;
- In the case of 'discontinuous' the point list is divided into consecutive sets of 4 points. Each set defines a single Bezier curve as defined below.

In the case of 'continuous' then after the first set of 4 points, defining the first Bezier curve, the subsequent curves are defined by three points each. The first point of each curve definition is omitted because it is identical to the last point of the preceding definition.

4.6.10.2.2 Geometric concepts. The following discussion assumes that there are 4 points defined for each Bezier curve. In one of the continuity conditions described above, one or two of the points may not actually be in the parameter data of the element. If the points in a given 4-point set are designated P_0, P_1, P_2, P_3 , then the defined Bezier curve goes from P_0 to P_3 using P_1 and P_2 as control points. The defined curve starts at P_0 and at P_0 is tangent to the line segment from P_0 to P_1 . The curve ends at P_3 and at P_3 is tangent to the line segment from P_2 to P_3 . The curve lies entirely within the convex hull defined by the points.

The curve is defined by the cubic parametric equations

$$\begin{aligned} X(t) &= A_x t^3 + B_x t^2 + C_x t + X_0 \\ Y(t) &= A_y t^3 + B_y t^2 + C_y t + Y_0 \end{aligned}$$

as t ranges from 0 to 1. The six coefficients $A_x, B_x, C_x, A_y, B_y, C_y$ are defined by

$$\begin{aligned} X_1 &= X_0 + \frac{C_x}{3} \\ Y_1 &= Y_0 + \frac{C_y}{3} \\ X_2 &= X_1 + \frac{(C_x + B_x)}{3} \\ Y_2 &= Y_1 + \frac{(C_y + B_y)}{3} \\ X_3 &= X_0 + C_x + B_x + A_x \end{aligned}$$

$$Y_3 = Y_0 + C_y + B_y + A_y$$

4.6.11 Closed figures

4.6.11.1 Construction of closed figures

A closed figure is a fill type compound object which commences with a BEGIN FIGURE element, followed by an ordered sequence of line and fill primitives (and optionally attributes and NEW REGION elements), and followed by END FIGURE. Edge attribute values are associated with the edge portions of the closed figure and fill attribute values are associated with the complete graphic object. BEGIN FIGURE and END FIGURE elements are delimiter elements; NEW REGION is a control element. The entire fill object is considered as a single unit on interpretation.

4.6.11.1.1 Closure point. The first point of the first line primitive in a new region is the closure point for that region. On interpretation this closure point is retained for use in closing the region. When the region is closed (with a NEW REGION or END FIGURE element, or by a fill primitive which begins a new region) an implicit boundary portion from the last point of the last line primitive in the region to this closure point is added to the closed figure on interpretation, unless these points are already coincident.

4.6.11.1.2 Regions. A closed figure consists of one or more regions. A region has a closed boundary which may be concave, convex, or self intersecting. A region is formed either by invoking a fill primitive in between BEGIN FIGURE and END FIGURE elements which closes the last region and contributes one or more complete regions; by invoking NEW REGION to start new regions to be formed from line primitives; or by a final invocation of END FIGURE. A closed figure constructed from only line primitives without use of NEW REGION consists of a single region.

The NEW REGION element may occur at any time during the closed figure construction. If the current region is closed, the element is ignored on interpretation. If the current region is open, an implicit boundary portion is added from the last point of the last primitive to the current closure point unless CONNECTING EDGE has been invoked after the last line primitive, in which case, an explicit boundary portion and edge portion is added by the CONNECTING EDGE line primitive.

4.6.11.2 Boundaries and edges

The boundary of each region consists of a combination of implicit boundary portions and edge portions.

4.6.11.2.1 Explicit boundary portions. Explicit boundary portions and edge portions are those added by the inclusion of primitives during closed figure construction. These are generated in the following situations:

- For fill primitives other than POLYGON SET, the complete edge becomes an explicit boundary portion and edge portion in the closed figure.
- For line primitives, only those portions become explicit boundary portions and edge portions which would normally be drawn if an interpreter were rendering the line primitive independently in a metafile. In particular for DISJOINT POLYLINE, only the segments from the first point to the second point, from the third point to the fourth point, and so on, become explicit boundary portions and edge portions when incorporated into closed figures.
- When a CONNECTING EDGE primitive precedes an element in the closed figure definition sequence which would otherwise have resulted in the addition of an implicit boundary portion to the closed figure, either to close a region (including closing the closed figure itself) or to connect two line primitives, then that CONNECTING EDGE primitive results in the portion added being an explicit boundary portion and edge portion. CONNECTING EDGE preceding or following DISJOINT POLYLINE or POLYGON SET does not affect the interpretation of those elements with respect to boundaries and edges.

Edge portions have associated edge attribute values taken from the current attribute values on interpretation. These values can be changed between the line and fill primitives that result in edge portions in a closed figure, and hence each edge portion has a distinct set of attribute values associated with it.

4.6.11.2.2 Implicit boundary portions. Edge attributes are never associated with implicit boundary portions. Implicit boundary portions are only rendered on interpretation for interior style HOLLOW and are a special representation of the interior, not a representation of any portion of the edge.

Implicit boundary portions are added on interpretation to the closed figure definition under the following circumstances:

- When NEW REGION, END FIGURE, or a fill primitive is interpreted and the current region has not been explicitly closed and CONNECTING EDGE has not occurred since the last line primitive, an implicit boundary portion is added from the last point of the last primitive to the current closure point to close the region.
- When the last point of the preceding line primitive is not coincident with the first point of the current line primitive, an implicit boundary portion is created to connect the last point of the preceding line primitive to the first point of the current line primitive.
- When portions of a DISJOINT POLYLINE primitive would not normally be rendered (i.e. from the second point to the third point, from the fourth point to the fifth point, and so on), implicit boundary portions are added between these points. (These are additional to the ones which may be added to connect to a preceding or following line primitive or to effect region closure after the disjoint polyline.)
- The portions of a POLYGON SET primitive as described below.

4.6.11.2.3 Conditions under which no boundary or edge is added. No boundary or edge portion is ever created connecting two regions, regardless of how those regions were created or closed.

4.6.11.3 Contribution of graphical primitive elements to the closed figure

4.6.11.3.1 Contribution of line elements to the closed figure. For line primitives, the 'first point' of a line primitive is connected to the 'last point' of the preceding line primitive, and the connecting implicit boundary portion becomes part of the boundary of the closed figure on interpretation. For each of the line primitives, the first and last points are defined to be as follows:

POLYLINE p1, p2, ..., pn: p1 is the first point; pn is the last point.

DISJOINT POLYLINE p1, p2, ..., pn: p1 is the first point; pn is the last point.

CIRCULAR ARC 3 POINT p1, p2, p3: p1 is the first point; p3 is the last point.

CIRCULAR ARC CENTRE,

CIRCULAR ARC CENTRE REVERSED:

The first point is the intersection of the circle with the ray (dx start, dy start) from the centre point (i.e. the clockwise end of the arc for CIRCULAR ARC CENTRE, the anticlockwise end of the arc for CIRCULAR ARC CENTRE REVERSED); the last point is the intersection of the circle with the ray (dx end, dy end) from the centre point (i.e. the anti-clockwise end of the arc for CIRCULAR ARC CENTRE, the clockwise end of the arc for CIRCULAR ARC CENTRE REVERSED).

ELLIPTICAL ARC:

The first point is the intersection of the ellipse with the ray (dx start, dy start) from the centre point; the last point is the intersection of the ellipse with the ray (dx end, dy

end) from the centre point.

For GDPs which generate line primitives, the first point is the first point of the point list; and the last point is the last point of the point list, as defined in the in the GDP registration and associated documentation.

The first point is the intersection of the hyperbola with the ray (dx_start, dy_start) from the centre point, and the last point is the intersection of the hyperbola with the ray (dx_end, dy_end) from the centre point.

The first point is the start point, the last point is the end point.

NON-UNIFORM B-SPLINE,
NON-UNIFORM RATIONAL B-SPLINE

The first point is the point of the curve corresponding to the start value of the parameter, and the last point is the point of the curve corresponding to the end value of the parameter.

POLYBEZIER

For the case 'continuous': the first point corresponds to the point defined by $t=0$ for the first 4 control points, and the last point corresponds to the point defined by $t=1$ for the last 3 control points. For the case of 'discontinuous': each 4 points define an individual Bezier curve, and an implicit boundary is drawn joining the last point of each Bezier curve to the first point of the following curve (similar to DISJOINT POLYLINE).

CONNECTING EDGE

If the region is open, the start point of the connecting edge is the last point of the last line primitive, and the end point of the connecting edge is either the first point of the following primitive or the current closure point as described above. If the connecting edge would be of zero length (i.e. if the two points it connects are coincident), the element is ignored on interpretation. The current modal values of the edge attributes are associated with any edge portion generated by this element.

If the current region is not open, invocations of the CONNECTING EDGE elements encountered are ignored on interpretation (i.e. CONNECTING EDGE shall not be used to connect regions).

Invoking CONNECTING EDGE multiple times after a line primitive results in the first instance (with its associated attributes) being used on interpretation.

On interpretation the theoretical definitions of the line primitives, not their renditions on the display surface, are used to define the explicit boundary portions of the closed figure. In particular, clipping does not apply to the construction of the closed figure, and the gaps or spaces of the edge type or the rendered width of the edge width do not affect the definition of the boundary of the closed figure.

4.6.11.3.2 Contribution of fill elements to the closed figure. Each fill primitive contributes a complete region to the figure (POLYGON SET may contribute more than one), after first closing the current region if one is open. On interpretation, an implicit NEW REGION is performed before and after a fill primitive (i.e.

Concepts

Graphical primitive elements

the new region resulting from a fill primitive is closed, and the next primitive begins a new region.)
The unclipped boundary of each fill primitive contributes to the unclipped boundary of the closed figure.

POLYGON SET primitives contribute to closed figure construction as follows:

- A POLYGON SET is considered to contribute one or more complete regions. If the current region has not been closed, an implicit NEW REGION is performed before the POLYGON SET is added to the figure definition. If the POLYGON SET does not end with a point whose edge-out flag is 'close visible' or 'close invisible', an implicit NEW REGION is performed after the POLYGON SET.
- Sequences of points with edge-out flag 'visible' are treated as if they were polylines, terminating with the first point with a different edge-out flag. Each such polyline becomes an edge portion of the boundary of the figure. The edge attribute values (including EDGE VISIBILITY) in effect when POLYGON SET occurs are associated on interpretation with any edge portion added in this way.
- Sequences of points with edge-out flag 'invisible' contribute implicit boundary portions which are polylines joining the points in the sequence, but not edges. Edge attribute values are not associated with these.
- Points with edge-out flag 'close invisible' generate the equivalent of a NEW REGION, generating an implicit boundary portion from this point to the current closure point if these are not coincident, and closing the current region.
- Points with edge-out flag 'close visible' generate the equivalent of a CONNECTING EDGE followed by a NEW REGION, resulting in an edge portion from this point to the current closure point if these are not coincident. The edge attribute values (including EDGE VISIBILITY) in effect when POLYGON SET is invoked are associated with any edge portion added in this way.

4.6.11.3.3 Contribution of GDPs to the closed figure. A GDP which is defined as a line primitive shall specify which is the first point and the last point in its point list, with respect to closed figure construction. Such GDPs are assumed to contribute to a closed figure a boundary corresponding to the unclipped locus which would be rendered on interpretation if the element occurred outside closed figure construction. Any other behaviour shall be as documented explicitly in the GDP description. A GDP which is defined as being a fill primitive is treated as described in the previous section. Any variation or special handling for closed figure construction shall be documented explicitly in the GDP description.

4.6.11.4 Examples of closed figures

Examples of closed figures are shown in figure 9.

The POLYGON SET example shown in figure 30 may also be obtained using the closed figure:

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
  POLYLINE (P3, P1, P2)
  NEW REGION (see note 1)
  POLYLINE (P4, P5, P6, P4)
END FIGURE
```

NOTE 1 Invisible implicit boundary portion P2-P3 generated.

Figure 9a shows the closed figure resulting from interpretation of the elements listed below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
  POLYLINE (P1, P2)
  CIRCULAR ARC 3 POINT (P2, P3, P4)
```

Graphical primitive elements

Concepts

```
POLYLINE (P4, P5)
CIRCULAR ARC 3 POINT (P5, P6, P1)
END FIGURE
```

Figure 9a could also be the result of interpreting the following sequence of elements which include CONNECTING EDGE.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
  CIRCULAR ARC 3 POINT (P2, P3, P4)
  CONNECTING EDGE
  CIRCULAR ARC 3 POINT (P5, P6, P1) (see note 2)
  CONNECTING EDGE
END FIGURE (see note 3)
```

NOTES

- 2 Visible edge portion P4..P5 generated.
- 3 Visible edge portion P1..P2 generated.

Figure 9b shows the closed figure resulting from interpretation of the elements listed below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
  POLYLINE (P1, P2, P3, P4)
  CIRCULAR ARC 3 POINT (P4, P5, P1)
  EDGE VISIBILITY (OFF)
  NEW REGION
  CIRCULAR ARC CENTRE (P7, 1, 0, 1, 0, |P7 - P5|) (See Note 4)
END FIGURE
```

NOTE 4 P7 is the mid-point of the line segment between P5 and P6.

Figure 9c shows the closed figure resulting from interpretation of the elements listed below.

```
BEGIN FIGURE
  CIRCULAR ARC CENTRE (P1, 1, 0, 1, 0, |P3 - P1|)
  NEW REGION
  CIRCULAR CENTRE (P1, 1, 0, 1, 0, |P2 - P1|)
END FIGURE
```

Figure 9c could also be the result of interpreting the following sequence of elements which include fill area elements.

```
BEGIN FIGURE
  CIRCLE (P1, |P3 - P1|)
  CIRCLE (P1, |P2 - P1|)
END FIGURE
```

Figure 9d shows the use of ELLIPTICAL ARC to draw a box with rounded corners and is the result of interpreting the sequence of elements shown below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
  ELLIPTICAL ARC (P1, P2, P3, (1,0), (0,1))
  CONNECTING EDGE
  ELLIPTICAL ARC (P4, P5, P6, (0,1), (-1,0)) (see note 5)
  CONNECTING EDGE
```

ELLIPTICAL ARC (P7, P8, P9, (-1,0), (0,-1))
CONNECTING EDGE
ELLIPTICAL ARC (P10, P11, P12, (0,-1), (1,0))
CONNECTING EDGE
END FIGURE (see note 6)

NOTES

- 5 Visible edge portion P2..P5 generated; edge portions P6..P8 and P9..P11 are drawn with the next two arcs.
- 6 Visible edge portion P12..P3 generated.
- Figure 9e shows the use of CIRCULAR ARC 3 POINT to create an 'S' shape and is the result of interpreting the sequence of elements shown below.

EDGE VISIBILITY (ON)
BEGIN FIGURE
CIRCULAR ARC 3 POINT (P1, P2, P3)
CIRCULAR ARC 3 POINT (P3, P4, P5)
CONNECTING EDGE
CIRCULAR ARC 3 POINT (P6, P7, P8) (see note 7)
CIRCULAR ARC 3 POINT (P8, P9, P10)
CONNECTING EDGE
END FIGURE (see note 8)

NOTES

- 7 Visible edge portion P5..P6 generated.
- 8 Visible edge portion P10..P1 generated.

Figure 9f shows the closed figure resulting from interpretation of the elements listed below. It is similar to figure 9d, but makes use of changing the edge attributes between successive occurrences of CONNECTING EDGE.

EDGE VISIBILITY (ON)
BEGIN FIGURE
ELLIPTICAL ARC (P1, P2, P3, (1,0), (0,1))
CONNECTING EDGE
EDGE TYPE(SOLID)
ELLIPTICAL ARC (P4, P5, P6, (0,1), (-1,0)) (see note 9)
EDGE TYPE(DASHED)
CONNECTING EDGE
ELLIPTICAL ARC (P7, P8, P9, (-1,0), (0,-1)) (see note 10)
EDGE TYPE(SOLID)
CONNECTING EDGE
ELLIPTICAL ARC (P10, P11, P12, (0,-1), (1,0))
EDGE TYPE(DASHED)
CONNECTING EDGE
END FIGURE (see note 11)

NOTES

- 9 No edge portion P2..P5 generated.
- 10 Visible (dashed) edge portion P6..P8 generated; solid edge portion P9..P11 drawn with the next arc.
- 11 Visible (dashed) edge portion P12..P3 generated.

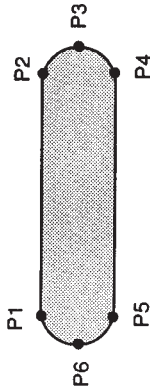


Figure 9a

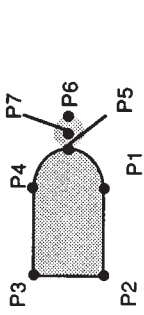


Figure 9b

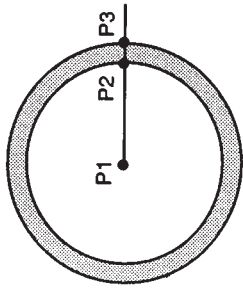


Figure 9c

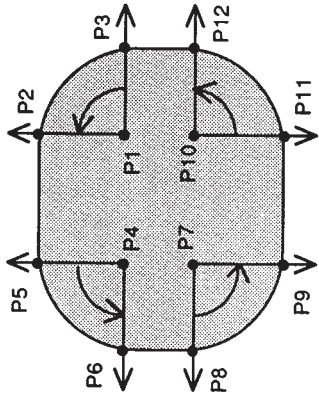


Figure 9d

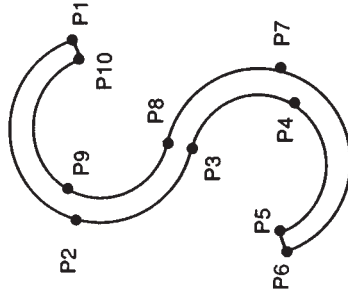


Figure 9e

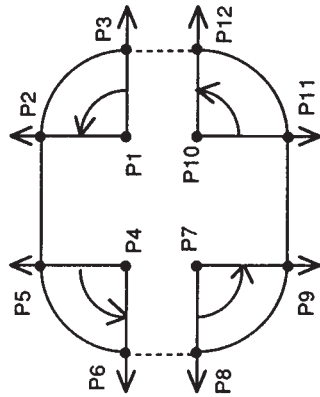


Figure 9f

Figure 9 — Examples of closed figures

4.6.12 Symbol elements

4.6.12.1 Description

This part of ISO/IEC 8632 defines mechanisms to access external symbol libraries and include their symbols in the metafile by reference. There is one symbol primitive element.

POLYSYMBOL generates a symbol which will be sized and oriented according to the symbol attributes and placed with its reference point coinciding with each point in a specified list of position points.

4.6.12.2 Attributes

The selection, sizing and placement of symbols is specified by the attribute elements SYMBOL SIZE, SYMBOL COLOUR, SYMBOL ORIENTATION, and SYMBOL LIBRARY INDEX. These are further described in 4.7.9.

4.6.12.3 Transformation of symbol elements

The vectors specified by the SYMBOL ORIENTATION element (see 4.7.9) are subject to the VDC-to-Device mapping (see 4.4.7) as well as to both segment and copy transformation (see 4.10.4.2 and 4.10.5).

4.7 Attribute elements

Attribute elements determine the appearance of graphical primitive elements. Attributes are classified as either individual attributes or attributes that may be bundled. Table 2 lists the attributes by this classification.

Bundled selection of attributes implies that the appearances of graphical primitive elements are distinguishable from one another when different bundles are specified. The method of specification of the aspects of a graphical primitive element that may be bundled may be chosen separately for each aspect. A further group of attributes called ASPECT SOURCE FLAGS (ASFs) takes the one of values 'individual' or 'bundled' to specify the choice. There is one ASF for each aspect that may be bundled.

There is a current modal value for every attribute. Elements are provided to change these modal values. The modal value established by setting an attribute remains until it is explicitly changed. All attributes return to their default values when the BEGIN PICTURE element is encountered.

There is at least one bundle index associated with each of the graphical primitive element types — line, marker, filled area, and text. Line, marker, and text elements have a single associated bundle index. Filled-area elements have two associated bundle indexes, one for interior attributes and one for edge attributes.

The value of each bundle index attribute is modally bound to subsequent graphical primitive elements of the associated type. Distinct values of the bundle index correspond to distinct appearances of the graphical primitive element.

For individual attributes, the current modal value is used to display a graphical primitive element. For attributes that may be bundled a graphical primitive element is displayed as follows:

- if the ASF for an aspect is 'individual', the value used is the current modal value (which is set only by the individual aspect-setting elements);
- if the ASF for an aspect is 'bundled', the value used is obtained via the bundle table for that primitive; the corresponding component of the bundle, which is pointed to by the bundle index, is used.

Table 2 — Individual attributes and attributes that may be bundled

Individual	May Be Bundled
CHARACTER HEIGHT	LINE TYPE
CHARACTER ORIENTATION	LINE WIDTH
TEXT PATH	LINE COLOUR
TEXT ALIGNMENT	MARKER TYPE
CHARACTER SET INDEX	MARKER SIZE
ALTERNATE CHARACTER SET INDEX	MARKER COLOUR
EDGE VISIBILITY	TEXT FONT INDEX
FILL REFERENCE POINT	TEXT PRECISION
PATTERN SIZE	CHARACTER EXPANSION FACTOR
PICK IDENTIFIER	CHARACTER SPACING
LINE CAP	TEXT COLOUR
LINE JOIN	INTERIOR STYLE
LINE TYPE CONTINUATION	FILL COLOUR
LINE TYPE INITIAL OFFSET	HATCH INDEX
TEXT SCORE TYPE	PATTERN INDEX
RESTRICTED TEXT TYPE	EDGE TYPE
INTERPOLATED INTERIOR	EDGE WIDTH
EDGE CAP	EDGE COLOUR
EDGE JOIN	
EDGE TYPE CONTINUATION	
EDGE TYPE INITIAL OFFSET	
SYMBOL LIBRARY INDEX	
SYMBOL COLOUR	
SYMBOL SIZE	
SYMBOL ORIENTATION	

The resulting appearance is interpreter dependent, but the intent is that the interpreter render distinct appearances of graphical primitive elements for distinct values of the associated bundle index (or indexes) by manipulation of the attributes that may be bundled. For example, LINE BUNDLE INDEX designates visually distinct combinations of the polyline attributes LINE WIDTH, LINE TYPE, and LINE COLOUR. Table 3 lists the aspects of each bundle.

Because inquiry of bundle representations is not generally possible in a metafile environment, mixing of 'individual' and 'bundled' ASF values within a bundle will compromise the guarantee of distinguishability of different bundle indexes within that bundle at interpretation time.

In addition to the attribute elements listed in table 3, there are elements permitting the definition of "compound attributes" from several of the graphical primitive elements. The following compound attribute is defined: compound text path. The elements that may be used to specify this compound attribute are listed in table 4.

Table 3 — Aspects of the bundle and affected primitives

Bundle	Aspects	Affected primitives
LINE	LINE TYPE	POLYLINE
	LINE WIDTH	DISJOINT POLYLINE
	LINE COLOUR	CIRCULAR ARC 3 POINT CIRCULAR ARC CENTRE ELLIPTICAL ARC
		CIRCULAR ARC CENTRE REVERSED
		CONNECTING EDGE HYPERBOLIC ARC PARABOLIC ARC
MARKER	MARKER TYPE	NON-UNIFORM B-SPLINE
	MARKER SIZE	POLYBEZIER
FILL	MARKER COLOUR	POLYMARKER
EDGE	INTERIOR STYLE	POLYGON
	FILL COLOUR	POLYGON SET
	HATCH INDEX	RECTANGLE
	PATTERN INDEX	CIRCLE
		CIRCULAR ARC 3 POINT CLOSE CIRCULAR ARC CENTRE CLOSE ELLIPSE ELLIPTICAL ARC CLOSE
TEXT	EDGE TYPE	POLYGON
	EDGE WIDTH	POLYGON SET
	EDGE COLOUR	RECTANGLE
		CIRCLE
		CIRCULAR ARC 3 POINT CLOSE CIRCULAR ARC CENTRE CLOSE ELLIPSE ELLIPTICAL ARC CLOSE
TEXT	TEXT FONT INDEX	TEXT
	TEXT PRECISION	RESTRICTED TEXT
	CHARACTER EXPANSION FACTOR	APPEND TEXT
	CHARACTER SPACING	TEXT COLOUR

Table 4 — Contributing primitives to compound attributes

Compound Attribute	First Element	Primitives Included	Other Elements	Final Element
Compound	BEGIN COMPOUND	Line primitives (note 1)		END COMPOUND
Text Path	TEXT PATH	GDP (note 2)	none	TEXT PATH

NOTES

- 1 All primitives of the identified classes may be included.
- 2 Whether a GDP element may contribute to the compound text path and whether or how it specifies that the compound path state be opened, maintained, or closed is specified with the definition of the GDP in the International Register of Graphical Items.

4.7.1 Line attributes

4.7.1.1 Line bundle

The LINE BUNDLE INDEX selects one entry in a table of bundled attribute values. The following attributes are in this bundle:

- LINE TYPE: determines the type of the line (for example, 'dotted', 'dashed', etc.) with which the line is rendered;
- LINE WIDTH: determines the width of the line with which the line is rendered;
- LINE COLOUR: determines the colour in which the line is rendered.

4.7.1.2 Individual line attributes

In addition to the line attributes which may be bundled there are a number of individual line attributes.

- LINE CAP specifies the appearance of the endpoints of line elements as well as the endpoints of individual dashes when dashed lines are rendered. The following cap styles are supported:

unspecified: no specific treatment is required;

butt: the line is squared off at the endpoint, there is no projection beyond the endpoint;

round: a semicircular arc with diameter equal to the line width is drawn around the endpoint and filled in (the drawn line thus projects beyond the endpoint);

projecting square: the line is squared off at a distance equal to half the line width beyond the endpoint;

triangle: a cap is added to the line which is an equilateral triangle, the length of whose side equals the line width;

NOTE 1 The LINE CAP element is only permissible in Version 3 metafiles, therefore only the 'unspecified' style is available in Version 1 and Version 2 metafiles.

These styles may be applied to the open endpoints of line elements — those endpoints which demark the beginning or ending of the entire line primitive — or to interior endpoints, which correspond to the endpoints of individual dashes when a non-solid line type is in effect. The interior caps must either match the caps on the open endpoints, have butt style, or be interpreter dependent. Figure 10 illustrates the styles of LINE CAP.

- LINE JOIN specifies the appearance of the interior corners of line elements. Interior corners correspond either to the interior vertices of polyline elements or to the junctions between distinct elements comprising a compound line element. The following styles are supported:

unspecified: no specific treatment is required;

mitre: the outer edges of the two adjoining line segments are extended until they meet at a point;

Concepts	Attribute elements
round:	a circular arc with diameter equal to the line width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner;
bevel:	the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

NOTE 2 The LINE JOIN element is only permissible in Version 3 metafiles, therefore only the 'unspecified' style is available in Version 1 and Version 2 metafiles.

For the style 'mitre', the rendering of the line join is affected by the MITRE LIMIT Control Element. Figure 10 illustrates the defined styles of LINE JOIN.

Both line caps and line joins behave as does line width with respect to transformation. If the value of LINE WIDTH SPECIFICATION MODE is *absolute* then conceptually the line cap and join are applied to the line in VDC space before any transformations are applied and they are subject to all transformations associated with the line. Otherwise, they are applied to the line in device space, conceptually after all associated transformations have been applied, and are immune to all transformations.

- c) LINE TYPE CONTINUATION provides control of the behaviour of non-solid line types at interior vertices and junctions of line elements. The following behaviours may be selected:

unspecified:	no specific treatment is required;
continue:	the style is continued without interruption across vertices;
restart:	the style is restarted at each vertex;
adaptive continue:	the style is continued, but each vertex must be "inked" including vertices at the ends of the line primitive which might otherwise not be drawn because of a non-solid line type.

NOTE 3 The LINE TYPE CONTINUATION element is only permissible in Version 3 metafiles, therefore only the 'unspecified' style is available in Version 1 and Version 2 metafiles.

- d) LINE TYPE INITIAL OFFSET allows control of how much of the first cycle of a non-solid line type to omit before drawing commences for a line primitive. It is specified as a fraction of one full cycle.

The Picture Descriptor element LINE AND EDGE TYPE DEFINITION allows the precise definition of the solid/gap sequences which comprise a line or edge type. A definition is associated with an index by this element, and this index may be referred to within the picture by LINE TYPE and EDGE TYPE elements.

4.7.2 Marker attributes

4.7.2.1 Marker bundle

The MARKER BUNDLE INDEX selects one entry in a table of bundled attribute values. The following attributes are in this bundle:

- a) MARKER TYPE: determines the marker symbol that is drawn at the marker position (for example, 'dot', 'plus', etc.);
- b) MARKER SIZE: determines the size of the marker symbol;
- c) MARKER COLOUR: determines the colour in which the marker symbol is drawn.

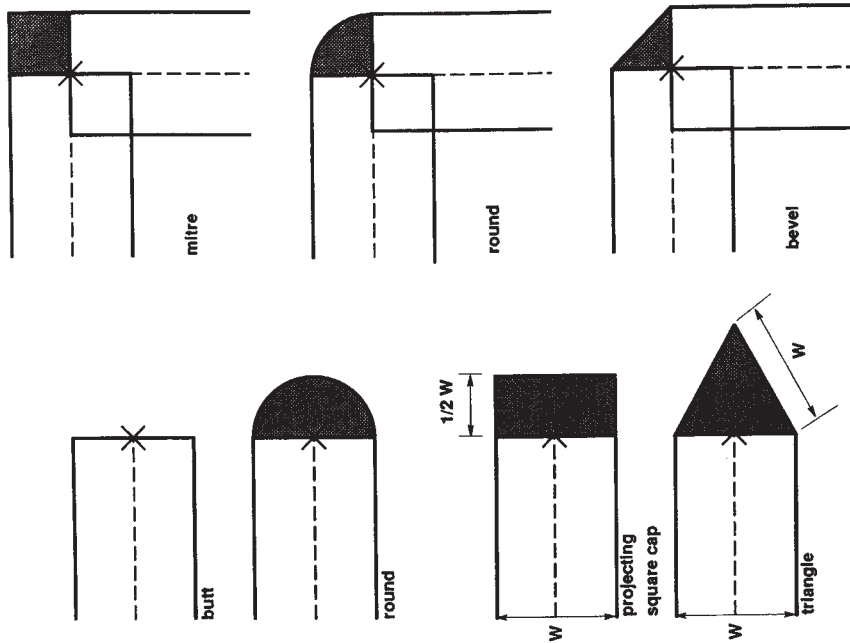


Figure 10 — Examples of LINE CAP and LINE JOIN

4.7.2.2 Individual marker attributes

There are no individual marker attributes in metafiles of Versions 1, 2, and 3 — all marker attributes are bundled.

4.7.3 Text attributes

4.7.3.1 Text bundle

The TEXT BUNDLE INDEX selects one entry in a table of bundled attribute values. The following attributes are in this bundle:

- TEXT FONT INDEX: determines the style of the graphical display of the text characters;
- TEXT PRECISION: determines the fidelity with which characters need be displayed and positioned;
- CHARACTER EXPANSION FACTOR: determines the deviation of the character width/height ratio from the ratio established by the font designer;
- CHARACTER SPACING: determines the amount of blank space added between characters in a string;
- TEXT COLOUR: determines the colour in which the text characters are drawn.

4.7.3.2 Individual text attributes and usage of text attributes

The representation and placement of text characters on a device is controlled by the attribute elements TEXT FONT INDEX, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, TEXT SCORE TYPE, and by the control elements AUXILIARY COLOUR and TRANSPARENCY. The placement and orientation of text strings is controlled by the attribute elements CHARACTER ORIENTATION, TEXT PATH, TEXT ALIGNMENT, RESTRICTED TEXT TYPE, and by the compound text path and the control element GENERALIZED TEXT PATH MODE. TEXT BUNDLE INDEX is an index into the text bundle table, each entry of which contains values for the attributes that may be bundled. Although the placement and size of text can be precisely specified by the attributes mentioned, the fidelity of rendering depends on the current TEXT PRECISION.

The choice of character font (that is, the style of the characters to be displayed) is determined independently of the character set. However, the specified font will only have meaning if it is related to the character set being used. Times Roman and Courier Bold Oblique are examples of commonly used fonts for Latin-based alphabets.

The attributes in the character representation and placement group (above) and TEXT BUNDLE INDEX may be changed within a string. A TEXT element or RESTRICTED TEXT element is tagged to show whether or not it is complete. If not complete, the element provides only the first portion of the string. The TEXT element or RESTRICTED TEXT element may be followed by the desired text attribute element(s) and then by an APPEND TEXT element, which provides the next portion of the string. This may be repeated as often as necessary, with the final APPEND TEXT tagged to indicate that the string is complete. Note that a metafile interpreter generally cannot display any of the text until the string is complete because of TEXT ALIGNMENT and the way in which attribute changes affect the definition of the text extent rectangle (see below). Text may be displayed before the string is complete only in the cases shown in table 5.

GENERALIZED PATH TEXT MODE has the possible values 'off', 'non-tangential', and 'axis-tangential'. If the mode is 'off', then the writing direction will be as specified by the TEXT PATH element — 'right', 'left', 'up', or 'down'.

Table 5 — Cases allowing display of partial text

Path	Vertical Alignment	Horizontal Alignment
right	normal vertical or baseline	normal horizontal, left, or continuous (0,0)
left	normal vertical or baseline	normal horizontal, right, or continuous (1,0)
down	top, capline, normal vertical, or continuous (0,1)	normal horizontal or centre
up	baseline, bottom, normal vertical, or continuous (0,0)	normal horizontal or centre

The four values of TEXT PATH define four special cases of the generalized text path, with paths which are straight lines pointing from the text position point in four indicated directions. For any of the four values of TEXT PATH when the generalized text path mode is 'off', the same effect can be achieved with a straight path and appropriate attribute settings in 'axis-tangential' mode.

When GENERALIZED TEXT PATH MODE is 'non-tangential' or 'axis-tangential', the string is laid out along the current text path as specified between the preceding BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH elements and positioned at the text position point according to the TEXT ALIGNMENT. The orientation of the characters along the path will depend on the mode. If the mode is 'non-tangential', the characters are positioned along the path and oriented as indicated by the character orientation vectors — each character has the same orientation regardless of the path direction. If the mode is 'axis-tangential', the characters are positioned along the path and for each character the character orientation vectors are rotated by the angle defined by the tangent to the path at the character's position — the orientation of each character depends upon the path direction at the character's placement point. In all cases, the angle between the up and base vectors of CHARACTER ORIENTATION is preserved and both the angle and the ratio of the vector lengths have the same effect as for mode 'off' (see figure 21, figure 22, and figure 23).

When the GENERALIZED TEXT PATH MODE is 'off' the escapement between characters will depend on TEXT PATH and will be computed by adding the width of the character to the character spacing if TEXT PATH is 'left' or 'right' and by adding the distance between the top and bottom lines of the character to the character spacing if the TEXT PATH is 'up' or 'down'.

When the GENERALIZED TEXT PATH MODE is 'non-tangential', or 'axis-tangential' the escapement will be computed by adding the width of the character to the character spacing. This escapement will be along the compound text path.

The characters are sized according to the CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR and are oriented according to CHARACTER ORIENTATION. The direction of character placement in the string relative to CHARACTER ORIENTATION is along the path defined within the scope of the preceding compound path definition. If the string length exceeds the length of the path, the characters of the string will continue to be placed along the path defined by a vector whose tail is the last point of the path and whose direction is the direction of the path at the last point.

TEXT ALIGNMENT is applied as follows for generalized text path. Conceptually the text is positioned along the defined path, determining the text alignment relative to the defined path, and applying other relevant text attributes. The path/text assembly is then aligned with the position point of the text element.

This is illustrated in figure 24 and figure 25. A "text extent band" is defined whose length is the larger of the arc length of the defined path and the length of the text as it has been placed along the defined path, and whose height in the direction of the character up vector has the character body height and alignment reference points defined in figure 11. This tube is then aligned relative to the text position point.

NOTE 2 If anisotropic transformations are in effect, then the height of the text will change for paths other than straight lines.

There are several methods for inclusion of characters from different character sets within a string. The method used is determined by the CHARACTER CODING ANNOUNCER Metafile Descriptor element. The default or normal technique is to use the CHARACTER SET INDEX element and restrict the contents of the text strings to printing characters and spaces (format effector control codes such as CR and LF are permitted, but their interpretation is implementation dependent). Other settings of the CHARACTER CODING ANNOUNCER or use of the ALTERNATE CHARACTER SET INDEX element permit standardized use of 8-bit characters and the SI, SO, and ESC control codes within the text string, in accordance with ISO 2022.

NOTE 3 SI and SO are only defined and usable in 7-bit coding. The ALTERNATE CHARACTER SET INDEX element is used to select a character set to be used as both the G1 set and the G2 set. The G1 set is used both for 8-bit characters in columns 10 to 15 of the code table, and with the SO control code. The assignment of meaning to the index parameter of both CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX is done with the CHARACTER SET LIST Metafile Descriptor element.

Selection of fonts from font tables is done by the TEXT FONT INDEX element. The assignment of meaning to the index values of TEXT FONT INDEX is done with the Metafile Descriptor element FONT LIST.

The font coordinate system is illustrated in figure 11. The character body encloses all of the drawn parts (kerning excepted) of all characters in the font (that is, no descender extends lower than 'bottom', and no accent mark or oversized glyph extends higher than 'top'). The left and right edges of the character body may be defined on a per-character basis to accommodate variable widths and proportional spacing. It is expected that font designers will specify some fonts having kerns extending beyond the character body. The body exceeds the actual character width and height as necessary to provide adequate white space between characters, such that text is readable and adequately separated when adjacent character bodies are flush (that is, when CHARACTER SPACING is 0). The character body is defined in this way to permit alignment of multi-line text without overlaps in the metafile environment. The CHARACTER HEIGHT specifies the VDC distance between the capline and baseline of the font (see figure 11). The CHARACTER EXPANSION FACTOR specifies the deviation of the width to height ratio of the characters from the ratio indicated by the font designer (see figure 12). CHARACTER SPACING specifies how much additional space is to be inserted between two adjacent character bodies (see figure 13). If the value of CHARACTER SPACING is zero, the character bodies are arranged one after the other along the TEXT PATH with only the intercharacter spacing designated by the font designer. If the value of CHARACTER SPACING is positive, additional space is inserted between character bodies. If the value of CHARACTER SPACING is negative, adjacent character bodies overlap although the characters themselves might not. Character spacing is specified as a fraction of the CHARACTER HEIGHT.

CHARACTER ORIENTATION specifies the character up vector and base vector, which fix the orientation, skew, and distortion of the characters, and also determine the sense of 'right', 'left', 'up', and 'down' for TEXT PATH and TEXT ALIGNMENT (see figure 14).

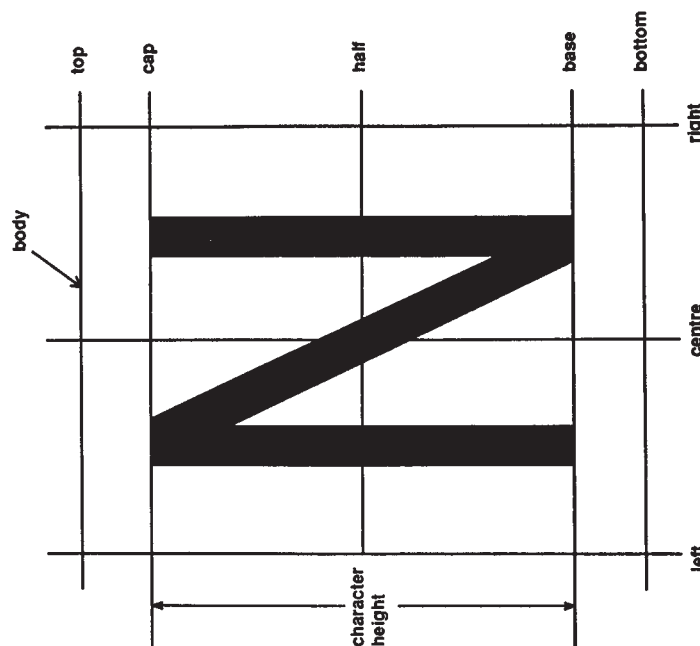


Figure 11 — Font description coordinate system

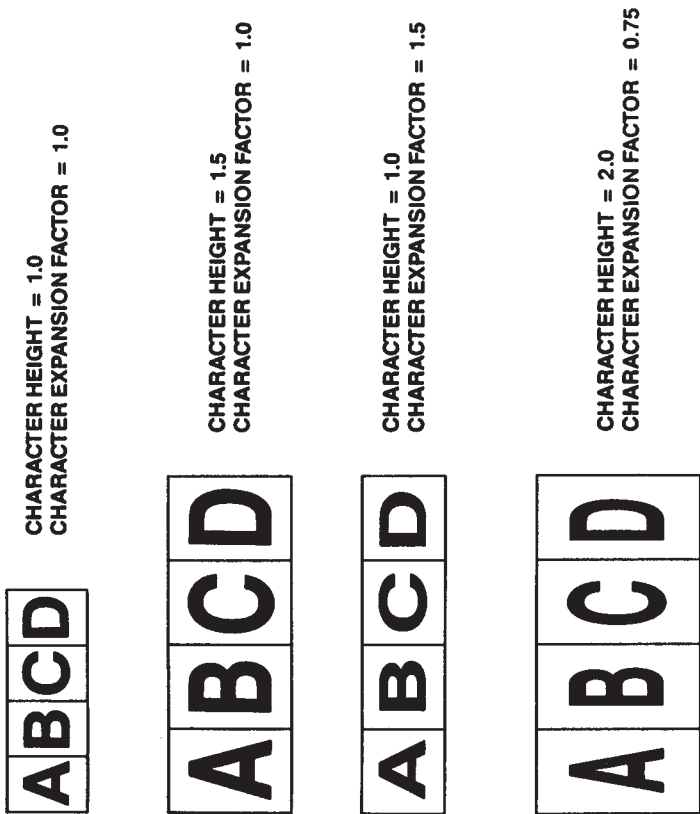


Figure 12 — CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR

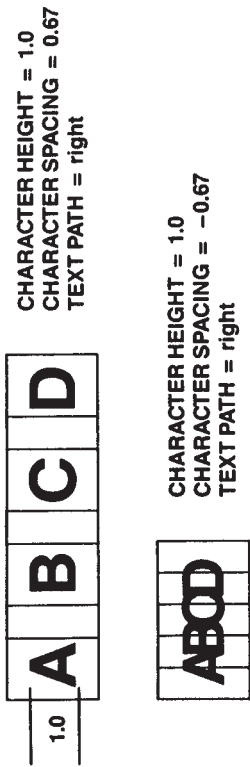


Figure 13 — CHARACTER SPACING

ISO/IEC 8632-1:1992 (E)
Attribute elements

Concepts

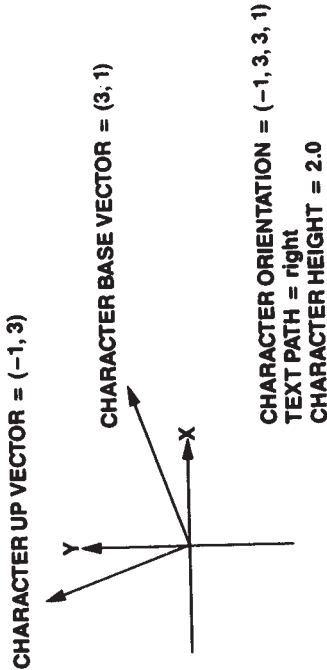
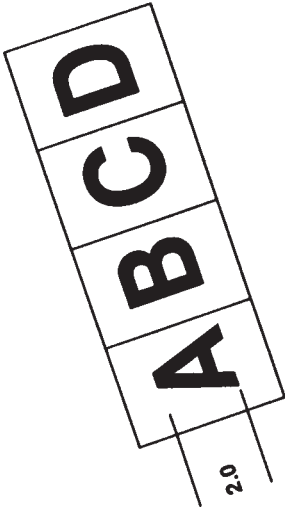


Figure 14 — CHARACTER ORIENTATION

ISO/IEC 8632-1:1992 (E)
Attribute elements

Concepts

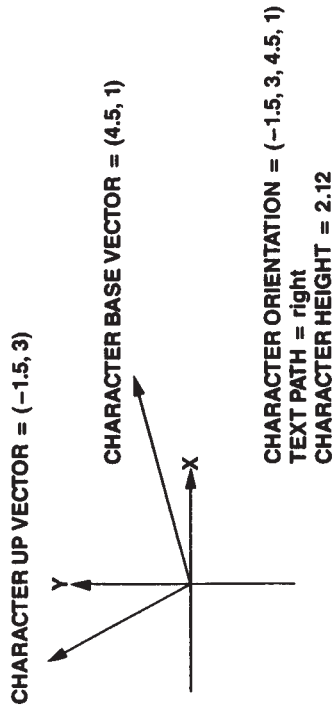
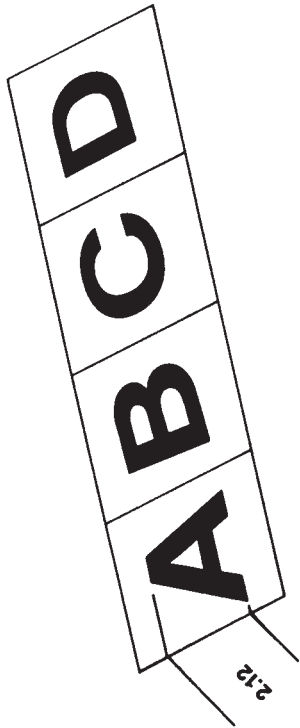


Figure 15 — CHARACTER HEIGHT and CHARACTER ORIENTATION after anisotropic transformation

Concepts

Attribute elements

The way in which software above the metafile generator and/or the metafile generator itself may use CHARACTER ORIENTATION is described. To generate the CHARACTER ORIENTATION and CHARACTER HEIGHT elements a vector whose length is the character height (baseline-to-capline) and whose direction is the desired character up vector is created. A second vector is also created with the same length, whose direction is negative 90° from the up vector. This pair of vectors may be transformed before being given to the metafile generator as the parameters to CHARACTER ORIENTATION. The length of the transformed up vector may then be used to generate the CHARACTER HEIGHT element. If an isotropic transformation is in effect above the metafile generator, the character height must be respecified by the metafile generator for each change in orientation (see figure 15). The CHARACTER HEIGHT and CHARACTER ORIENTATION are decoupled to permit changing character height (but not orientation) within a string. Thus, to the metafile interpreter the absolute lengths of the vectors in CHARACTER ORIENTATION are not significant; only their directions and the ratio of their lengths are significant.

The ratio of the length of the width vector to the length of the height vector is used to scale the CHARACTER SPACING for text paths 'right' and 'left', and the CHARACTER EXPANSION FACTOR in all cases, before these are used to display the text.

TEXT PATH has the possible values 'right', 'left', 'up', and 'down'. It specifies the writing direction of the text string as follows:

- right: means the direction of the character base vector;
- left: means 180° from the character base vector;
- up: means the direction of the character up vector;
- down: means 180° from the character up vector.

For the 'up' and 'down' text path directions, the characters are arranged so that the centres of the character bodies are on a straight line in the direction of the up vector of CHARACTER ORIENTATION. For the 'left' and 'right' text path directions, the characters are arranged so that the baselines of the characters are on a straight line parallel to the direction of the character base vector. These composition rules also hold true when characters of different heights, expansion factors, fonts, or precisions are intermixed in a string by means of attribute changes between non-final TEXT elements and subsequent APPEND TEXT elements.

Alignment of text is done with respect to a text extent rectangle, which is derived by joining the character bodies of the characters in the string according to the current status of the attributes and the composition rules described. Alignment is performed according to the highest precision in the string.

For TEXT PATH = 'left' or 'right',

- TOPLINE: topline farthest from the baseline
- CAPLINE: capline farthest from the baseline
- HALFLINE: halfline farthest from the baseline
- BOTTOMLINE: bottomline farthest from the baseline
- LEFT: leftmost edge of leftmost character body
- RIGHT: rightmost edge of rightmost character body
- CENTRE: halfway between left and right edges

For TEXT PATH = 'up' or 'down',

- TOPLINE: topline of topmost character

Attribute elements

- CAPLINE: capline of topmost character
- HALFLINE: halfway between halflines of topmost and bottommost character
- BASELINE: baseline of bottommost character
- BOTTOMLINE: bottomline of bottommost character
- LEFT: left edge farthest from the centreline
- RIGHT: right edge farthest from the centreline

Note that the relationship of topline to capline, bottomline to baseline, and the placement of the halfline are font-dependent (see figure 16). It is for this reason that the various defining lines of the text extent rectangle need not be derived from the same character body. This is a function of the text height, text font, text precision and character expansion factor changes within a string.

The TEXT ALIGNMENT attribute controls the positioning of the text extent rectangle in relation to the text position (see figure 17).

The horizontal component of TEXT ALIGNMENT has five possible values: 'left', 'centre', 'right', 'normal horizontal' and 'continuous horizontal'. If the horizontal component is 'left', the left side of the text extent rectangle passes through the text position. Similarly, if the value is 'right', the right side of the text extent rectangle passes through the text position. If the horizontal component is 'centre', the text position lies midway between the left and right sides of the text extent rectangle. In this case, if TEXT PATH = 'up' or 'down', the straight line passing through the centrelines of the characters also passes through the text position.

The vertical component of TEXT ALIGNMENT has seven possible values: 'top', 'cap', 'half', 'base', 'bottom', 'normal vertical', and 'continuous vertical'. A vertical alignment value of 'top', 'cap', 'half', 'base', or 'bottom' causes the text to be moved such that the corresponding defining line of the text extent rectangle passes through the text position.

For both horizontal and vertical alignment, normal values are converted to the appropriate value, as indicated in clause 5, at text element elaboration time and thereafter treated as above. For all values of TEXT ALIGNMENT the alignment value applies to the complete text string, which may be comprised of non-final partial strings and a final partial string.

If the value of the horizontal component of TEXT ALIGNMENT is 'continuous horizontal', an additional value, 'continuous horizontal alignment' (a real number normalized so that 1.0 corresponds to the width of the text extent rectangle) is used as an offset from the text position to the left side of the text extent rectangle. Figure 18 illustrates the sense of positive and negative values of 'continuous horizontal alignment'.

If the value of the vertical component of TEXT ALIGNMENT is 'continuous vertical', an additional value, 'continuous vertical alignment' (a real number normalized so that 1.0 corresponds to the height of the text extent rectangle) is used as an offset from the text position to the bottom side of the text extent rectangle. Figure 18 illustrates the sense of positive and negative values of 'continuous vertical alignment'.

The foregoing examples have been illustrated for the case of the character up vector and the character base vector being orthogonal. When they are not, the text extent rectangle becomes a parallelogram, with the sides remaining parallel to the two orientation vectors. The centreline skews to remain parallel with the left and right edges of the text extent parallelogram. The height of the text extent rectangle is measured along the skewed edge (not perpendicular to the baseline), and the distance to be moved for alignment is done along the angle made by the appropriate orientation vector (see figure 19). Right is in the direction of the character base vector, and left is in the opposite direction.