WebCGM 2.0

OASIS CGM Open WebCGM TC Committee Draft

17 March 2005

This Version:
http://TBD
Previous Version:
http://www.oasis-open.org/apps/org/workgroup/cgmo-webcgm/download.php/11362/webcgm20-D3.zip
Authors:
Benoit Bezaire, ITEDO
David Cruikshank, The Boeing Company
Lofton Henderson, HAL
Contributors
See Acknowledgements.

Abstract

CGM (Computer Graphics Metafile) has been an ISO standard for vector and composite vector/raster picture definition since 1987, and a registered MIME type since 1995. CGM has a significant following in technical illustration, electronic documentation, geophysical data visualization, amongst other application areas. WebCGM (1.0) -- first published in 1999 followed by a second (errata) release in 2001 -- represents an important interoperability agreement amongst major users and implementors of CGM, and thereby unifies potentially diverse approaches to CGM utilization in Web document application

WebCGM historically has been a joint effort of CGM Open Consortium, in collaboration with W3C graphics experts. Original work on WebCGM was supported by the European Commission Esprit project. While WebCGM is a binary file format, nevertheless WebCGM follows published W3C requirements for a scalable graphics format where such are applicable.

WebCGM includes a profile for the effective application of CGM in Web electronic documents.

The design criteria for the graphical content of WebCGM aim at a balance between graphical expressive power on the one hand, and simplicity and implementability on the other. A small but powerful set of metadata elements is standardized in WebCGM, to support the functionalities of hyperlinking and document navigation, picture structuring and layering, and, search and query on WebCGM picture content.

WebCGM 2.0 includes a DOM specification (API) for programmatic access to WebCGM objects, and a specification of an XML Companion File (XCF) architecture, for externalization of non-graphical metadata. WebCGM. In addition, WebCGM 2.0 builds upon and extends the graphical and intelligent content of WebCGM 1.0 -- functionality that was forecast in WebCGM 1.0, but was postponed in order to get the standard and its implementations to users quickly.

Status of This Document

This is the first publication of WebCGM 2.0 as an OASIS Committee Draft. The OASIS CGM Open WebCGM Technical Committee (TC) voted at its February face-to-face meeting to approve this specification as a Committee Draft and initiate public review. This specification has been developed so far within the WebCGM TC. The definitions of all WebCGM 2.0 components have reached "feature-freeze" within the WebCGM TC, and are in trial implementation now. If you wish to comment on this, please used the comment button on the WebCGM TC home page.

Contents

Detailed Table of Contents

Acknowledgements

In addition to the listed authors of this specification, the following individuals have contributed significantly to WebCGM development, to 1.0 and/or to 2.0:

Forrest Carpenter, System Development Inc.
Franck Duluc, Airbus
Stuart Galt, The Boeing Company
Ulrich Laesche, Ematek Informatik GMBH
Don Larson, Larson Software Technology
Kevin O'Kane, Auto-trol Technology
Dieter Weidenbrueck, ITEDO
Maryse Da-Ponte, Aerospatiale
Bruce Garner, Analyst/Consultant
Alan Hester, Xerox Corporation
Bob Hopgood, CCLRC
Chris Lilley, W3C
Brad Powell, Zeh Graphic Systems
Dave Rahnis, Bentley Systems
Lynne Rosenthal, NIST
John Gebhardt, Intercap Graphics Systems
Roy Platon, CCLRC
Ty Bartosh, Jeppesen Inc.

Up to cover page

CGMO specification - WebCGM 2.0 - Expanded Table of Contents

Contents


W3C
XHTML
1.0

MADE WITH
CASCADING
STYLE SHEETS

CGM Open specification - WebCGM 2.0 - Introduction to WebCGM

---

1. Introduction to WebCGM

This section is *informative*.

1.1 About WebCGM

The scope of this WebCGM™ 2.0 specification includes three components:

1. an intelligent graphics profile of the ISO Computer Graphics Metafile (CGM) standard (ISO/IEC 8632:1999), tailored to the requirements for scalable 2D vector graphics in electronic documents on the World Wide Web;
2. a WebCGM Document Object Model (DOM), which provides an application programming interface to WebCGM objects in WebCGM viewers;
3. definition of a standard WebCGM XML Companion File (XCF), which allows applications to externalize some non-graphical metadata from WebCGM instances, yet maintain a tight binding of the metadata to WebCGM objects.

WebCGM is a set of specifications targeted especially at the effective application of the ISO standard to representation of 2D graphical content within Web documents.

CGM has been an ISO standard since 1987, and CGM has been a registered media type (image/cgm) for the Internet and the World Wide Web since December 1995. WebCGM 1.0, comprising the original intelligent graphics profile of ISO CGM , was first published in 1999, was re-released in 2001 with error corrections, and forms the basis for the WebCGM 2.0 intelligent graphics profile of this WebCGM 2.0 specification.

1.2 The WebCGM profile and profile rules

The WebCGM profile is a conforming profile of ISO CGM under the stipulations of CGM:1999 Clause 9, "Profiles and conformance", and it utilizes the Profile Proforma (PPF) of CGM:1999 Annex I.1, Proforma tables, for representation of the element-by-element content details.

The WebCGM profile is an "intelligent graphics" profile, which means that in addition to graphical content based on CGM Versions 1-3, the profile includes non-graphical content

based on CGM Version 4, Application Structures. The non-graphical content allows the definition of hierarchies of application objects, as well as the association of metadata, such as link specifications and layer definitions, with the objects.

1.3 WebCGM requirements

The content of the original WebCGM 1.0 profile was chosen to satisfy the requirements articulated in the document "W3C Scalable Graphics Requirements", *http://www.w3.org/Graphics/ScalableReq* ([grfreq]), and the specifications and requirements defined in "Use of CGM as a Scalable Graphics Format", *http://www.w3.org/TR/NOTE-cgm* ([cgmreq]). The content of the WebCGM 1.0 profile was also significantly influenced by technical recommendations from the CGM Open consortium (see *http://www.cgmopen.org/*).

The selection criteria for the WebCGM profile include:

- graphical content: it should have high expressive power; and, it should be both widely implemented, and implementable with a reasonable level of effort.
- intelligence content (structuring and metadata elements): criteria came from the above-mentioned requirements document, [cgmreq], plus additional requirements generated during the first 5 years of deployment and use of the WebCGM 1.0 standard.

The upgraded content of the WebCGM 2.0 profile -- a set additions, deletions, and modification applied to the 1.0 profile -- has been shaped by:

- additional requirements generated during 5 years of deployment of WebCGM 1.0 in industry;
- apparent non-use of certain 1.0 features;
- need for convergence with similar profiles in closely related industries.

The DOM and XCF components of WebCGM 2.0 derive from requirements generated during 5 years of deployment of WebCGM 1.0 in industry. The WebCGM 2.0 Requirements document summarizes these requirements.

1.4 WebCGM and other profiles

The WebCGM 2.0 intelligent graphics profile, like its predecessor WebCGM 1.0, is a proflile of the ISO CGM:1999 standard, designed for effective application of CGM in technical Web applications. WebCGM is not aimed at or optimized for any particular technical application sector, but is intended to satisfy general requirements shared by different but closely related technical Web applications.

Following five years of deployment and application of WebCGM and other technical profiles (such as Air Transport Association's), some divergence began to appear. WebCGM 2.0 represents a major effort towards convergence of intelligent graphics profiles in closely related industries. In fact, it is the intention of the authors and publishers of WebCGM 2.0 that it be used as a basis for the definition of industry-specific profiles. The use of WebCGM as a core profile from which specific-industry technical profiles are derived and defined is defined in Cascading Profiles.

1.6 Editions and releases of WebCGM

CGM:1999 Clause 9, Profiles and conformance, prescribes that profiles shall maintain revision control by using a standard "ProfileEd" keyword. Instances of the profile carry this edition information in their identification section.

This release of WebCGM is Edition ("ProfileEd") 2.0. Higher editions may be defined at a later date.

This release is the first release of WebCGM 2.0. There may be future releases of WebCGM 2.0, for maintenance and defect correction.

1.7 Roadmap to this specification

WebCGM is written in these major sections:

- This section, containing introductory and overview material, which is *informative* but *not normative*.
- A WebCGM Concepts section, *informative* but *not normative*.
- Detailed descriptive material on the V4 content of WebCGM, including Content Model which can be used for V4 content. This section is *normative*.
- The definition of a standard XML Companion File (XCF) for use with WebCGM. This section is *normative*.
- The definition of the WebCGM Document Object Model (DOM). This section is *normative* (including normative IDL definitions of the DOM interfaces).
- The Profile Proforma (PPF), comprising an extensive table which addresses every element of the CGM standard, per CGM:1999 Annex I. This section is *normative*.
- Appendixes, including the normative Conformance appendix, and both normative and informative References sections.

Up to cover page | On to Concepts

---

2. WebCGM concepts

This chapter is informative (non-normative).

2.1 The structure of a WebCGM

A WebCGM is a Version 1, 2, 3, or 4 CGM as defined in ISO/IEC 8632:1999, with some restrictions. The restrictions improve the interoperability of WebCGM, and simplify the production of WebCGM interpreter (viewer) tools.

A WebCGM 2.0 instance, as shown in Figure 1, consists of a single Picture.

Properties which apply to the whole metafile are defined in the Metafile Descriptor. These include descriptive information about the metafile, the precisions of numbers, as well as identifiers for fonts and such resources. Properties which apply to the elements in the body of the picture are contained in the Picture Descriptor. These include such information as picture size and scaling, specification modes for aspects such as line width, and background color. Because WebCGM 2.0 allows only a single picture per metafile, the distinction -- whole-metafile versus picture-specific -- may not seem useful. However, because a WebCGM 2.0 metafile must be a valid ISO CGM:1999 metafile, the ISO CGM:1999 metafile structure is observed.

The WebCGM picture contains CGM graphic elements, as well as (optionally) Application Structures. Application Structures define intelligent objects within the picture, which are comprised of groups of graphical primitives. These intelligent objects may contain attributes or properties. WebCGM defines several types of intelligent objects - "graphical object", "paragraph", "layer", and "sub-paragraph" - as well as a few properties which each group may have.

WebCGM 2.0 also defines a purely graphical grouping mechanism, "graphical node", which groups graphical primitives as an Application Structure, but diallows the attributes or properties that associate intelligence with objects.

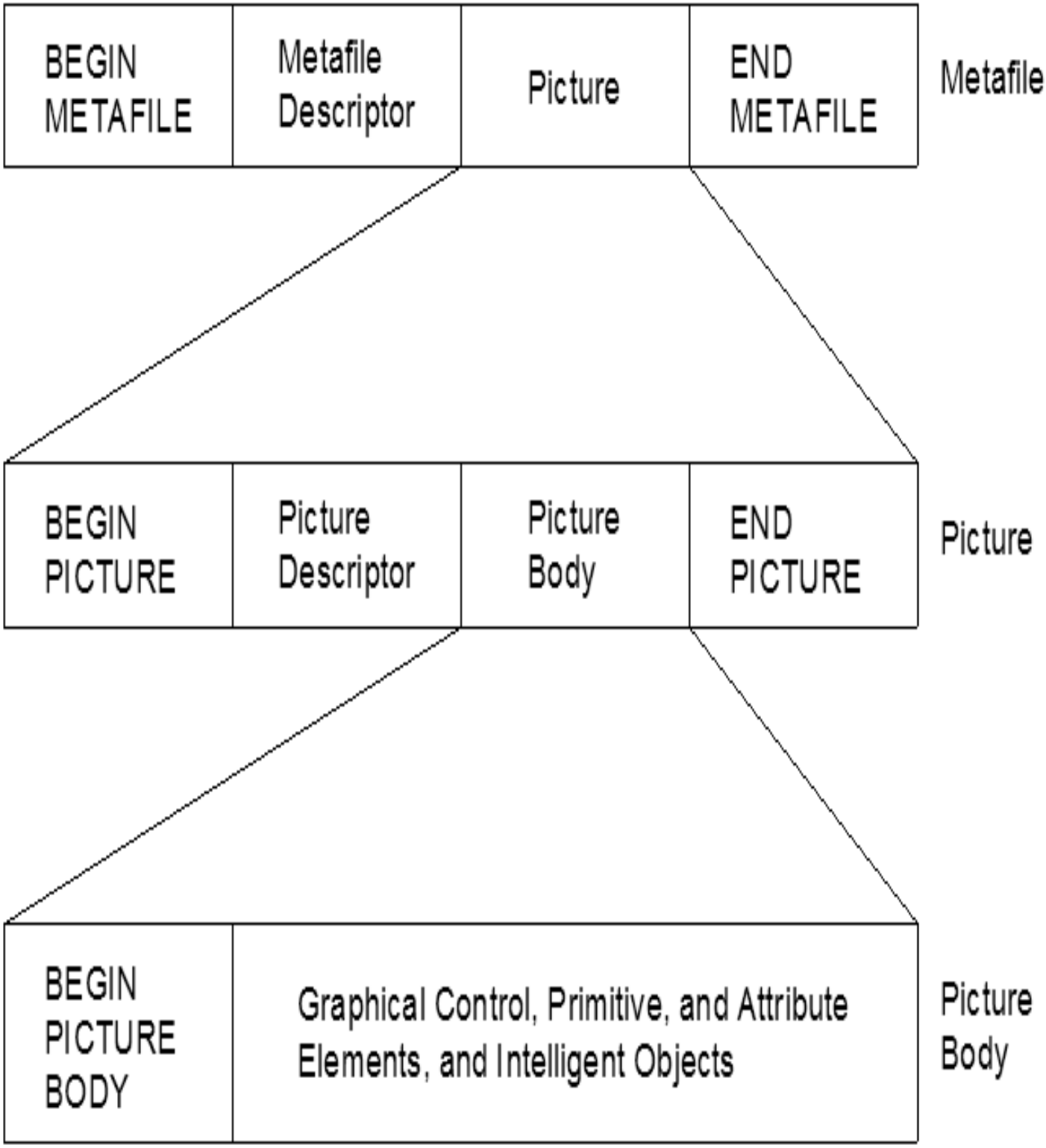


Figure 1. WebCGM File Structure

2.2 Picture content and usage

2.2.1 Raster and vector content

CGM supports both raster and vector graphics in the same picture. WebCGM permits the use of popular raster compression methods - CCITT group 4, JPEG, and the deflate (LZ77 derivative) method of PNG - for

raster content embedded within the picture. For information about scaling of WebCGM pictures in Web documents, see section WebCGM and the Object Tag.

2.2.2 Drawing model

Elements rendered first may be wholly or partially hidden by elements rendered later. In the ISO CGM standard, the writing mode of primitives is "replacement mode" - content is rendered opaquely on top of previous content. To meet per-primitive (and per-pixel) transparency requirements, WebCGM includes a registered extension for Alpha transparency, as well as registered colour models RGB-alpha and sRGB-alpha.

2.2.3 Overlaying a picture

In the ISO CGM:1999 standard, a picture has an implicit or explicit opaque background. Graphic elements within the picture are rendered in the order they appear in the metafile. It is a requirement of a 2D graphics format for Web documents that pictures may be overlayed on previous content. For this, it must be controllable whether the picture background is opaque or transparent (both cases are needed), or "translucent" (partially opaque).

This may be handled in two ways with WebCGM. First, the "TRANSPARENT" parameter of the OBJECT tag may be used (see section "WebCGM and the OBJECT Tag"). Second, the registered Alpha-transparency ESCAPE element may be included in the Picture Descriptor and applied to the background colour of the picture.

2.3 Intelligence - Objects, Layers, Hyperlinks, Metadata

2.3.1 Overview

Within a WebCGM picture, groups of graphical primitives can be defined which structure graphics to meet the requirements of integration into Web documents. Groups in WebCGM are realized as standard Version 4 Application Structures (APS) of ISO CGM.

To meet the requirements of intelligent graphics, four specific group types are defined and allowed in WebCGM: 'grobject', 'layer', 'para', and 'subpara'. WebCGM 2.0 allows a fifth group type, 'grnode', as a convenience for authoring tools to preserve their graphical grouping functions. The detailed normative syntax and semantics of the group types, including viewer behavior, is defined in Chapter 3 and in the PPF. Below is a brief conceptual summary.

Every group has at least one explicit property, its unique identifier (a parameter of the Begin APS element). WebCGM groups other than 'grnode' may have several explicit attributes associated with them. These attributes are realized as standard Version 4 Application Structure Attribute elements (APS Attributes) of ISO CGM.

WebCGM defines a detailed content model for the V4 content in the form of an "XML fragment". See section, "WebCGM Content Model".

2.3.2 WebCGM defined group types

WebCGM defines the following allowable group (APS) types, to support the Web document operations of hyperlinking, layered pictures, and text search within graphics:

- grobject - (graphical object) the basic grouping APS for identification of objects, principally used to identify sources and destinations of hyperlinks.
- layer - an APS type that allows the division of a picture into a set of graphical layers, for use by viewers in selective presentation and "2-1/2 D" effects.
- para - (paragraph) an APS type to facilitate text search within graphics, in cases such as multi-element, multi-line text, and other cases (e.g., polygonized text) where text search might otherwise be difficult (or impossible).
- subpara - may be used to identify smaller fragments of text within APS of type 'para', enabling, for example, the marking of the larger text block (the "paragraph") for searching purposes, and the tagging of smaller fragments as hotspots.

The detailed normative syntax and semantics are presented later in this profile. WebCGM 2.0 does allow one other group type for the convenience of authoring tools:

- grnode - may be used for simple grouping of graphical primitives, to preserve authoring tools' grouping functions; none of the facilities (properties & attributes) for attaching intelligence may be used with 'grnode'.

Note. 'Grnode' was not present in WebCGM 1.0, but has been added to WebCGM 2.0 to allow for better hierarchical structure in WebCGM documents. The 'grnode' APS (meant for 'group node') allows for preservation in the WebCGM metafile instance of the graphical grouping facilities that are typical in illustration authoring tools.

WebCGM does not allow private group types in WebCGM instances. External private metadata can be associated, by id or by name, with all group (APS) types other than 'grnode' within a WebCGM. A standard external mechanism is defined in the *XML Companion File* section.

2.3.3 Usage of WebCGM objects for navigation

Groups of types 'para', 'subpara', and 'grobject' may be used for picking and navigation operations in hyperlinked Web documents. These three APS types are called "objects" in WebCGM.

Objects may contain an explicit 'region' APS Attribute, which provides the boundary for picking operations. This is known as the *overlay model* of object identification (for picking and linking). It is useful in cases of badly structured legacy graphics, raster content, and in some cases to optimize picking operations.

Objects which contain graphical content have an implicit property: the boundary or bounding extent of the enclosed graphical object. This extent is used for picking and navigation operations in hyperlinked Web documents, in the absence of a 'region' attribute. Use of this implicit boundary property for picking and navigation operations in Version 4 CGM instances is referred to as the *embedded model* - a powerful model for newly authored Web documents.

Objects may also be the target of a link. Viewers will generally move the APS into view and scale them to fit into the viewer's rectangle. The exact viewer behavior is controlled by a set of *object behavior* keywords associated with the link, and the presence or absence of certain attributes on the object ('viewcontext', 'region', etc.)

2.3.4 WebCGM defined group properties

Explicit properties or attributes of WebCGM groups are encoded as APS Attribute elements. Each APS Attribute has a "type" parameter, which identifies the property or attribute. WebCGM defines the following allowable group APS Attribute types:

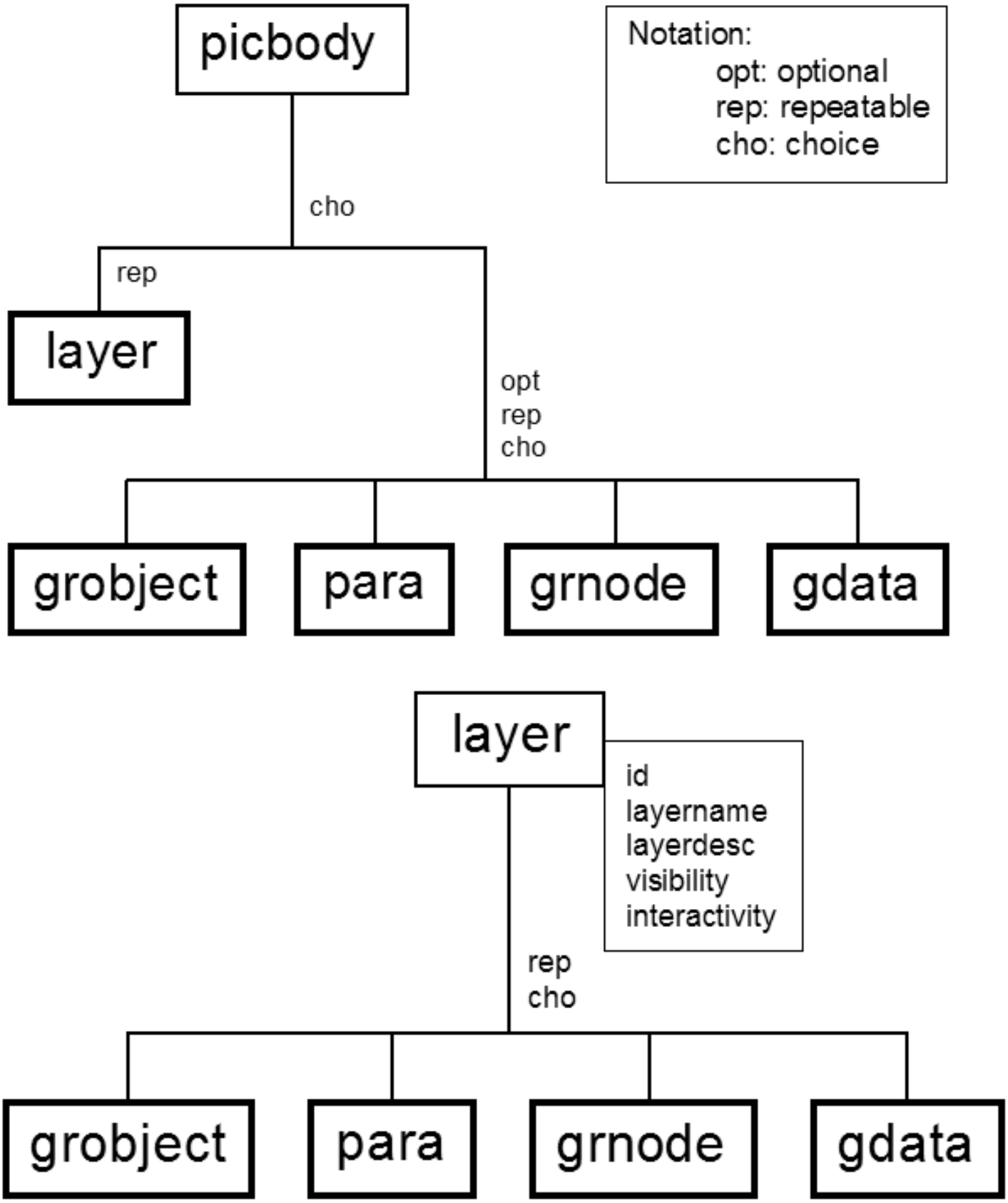
- region - defines a spatial region, to be (optionally) associated with an APS for picking and navigation purposes.
- viewcontext - defines a rectangle, to be associated with an APS for establishing the initial view upon execution of a link to that APS.
- linkuri - defines a link, whose target is specified by a URI, for hyperlinking to text content, pictures in other metafiles, or objects within the same or other metafiles. "#" fragment syntax is defined for a full addressing model down to the object level within the picture, and specifying viewer behavior upon link traversal.
- layername - the name (or number) to be assigned within an APS of type 'layer'.
- layerdesc - optional "layer description" string within an APS of type 'layer'.
- screentip - a string to be associated with an object, to be shown in the typical Web browser "screen tip" style when cursor passes over the object.
- name - a "common name" attribute to be associated with object, that gives a useful search handle or way of defining searchable subtypes of the object type; also allows a group of same-name objects to be a link target.
- content - an optional attribute of the 'para' and 'subpara' APS, which can give a reasonable basis for searching in the case of badly structured text within WebCGM instances.
- visibility -- the 'visibility' attribute indicates if an object is visible (drawn) or not, and also potentially disables its eligibility to be picked (invisible objects are ineligible for picking).
- interactivity -- the 'interactivity' attribute indicates whether or not an object is eligible to be picked (i.e., may receive mouse events), and also affects a handful of other interaction-related behaviors (see normative Chapter 3 description for details).

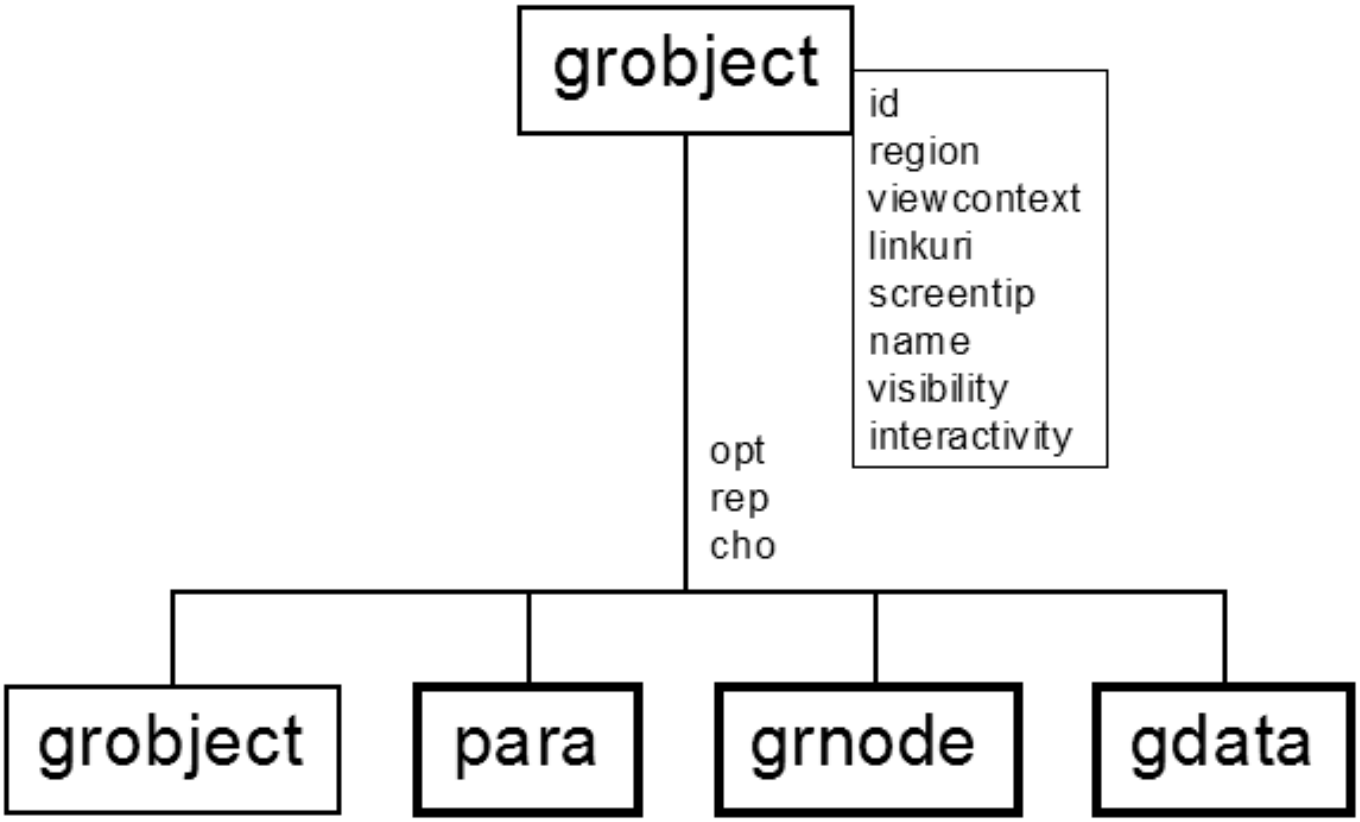
WebCGM does not allow private attribute types in WebCGM instances. External private metadata, including attributes and properties, can be associated by id or by name with all group (APS) types other than 'grnode' within a WebCGM. A standard external metadata binding mechanism is defined in the *XML Companion File* chapter .

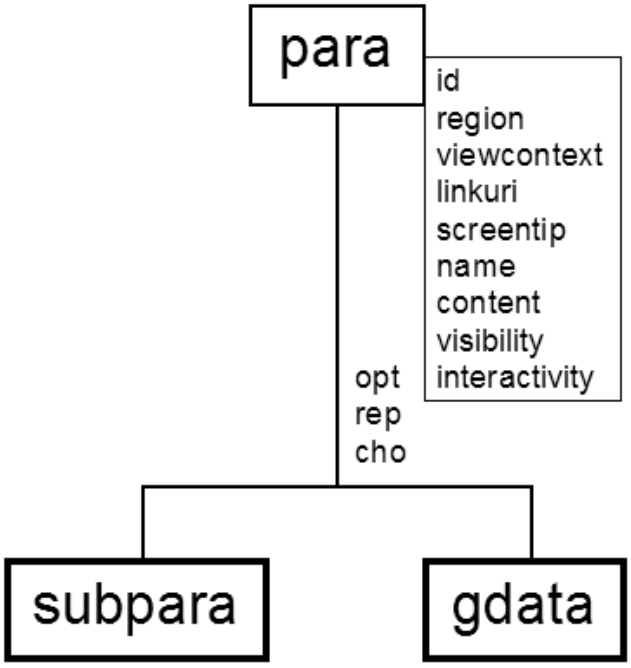
2.3.5 Content Model

The detailed normative syntax and semantics are presented later (Chapter 3) in this specification. The structure and relationships of the intelligence content are illustrated in the following diagrams. In the following, picbody is not a specific WebCGM object type, but rather a convenience to refer to that part of the CGM picture which is between the Begin Picture Body element and the End Picture Element, exclusive. Boxes with heavy borders indicate elements that are decomposed further, and offset boxes indicate attributes associated with an element. Similarly, gdata is not an object type, but rather a catch-all reference

to zero or more CGM graphical elements which WebCGM allows, and which are valid at such a position according to the rules of CGM. The cgmprim attribute associated with gdata represents an entity that associates the graphical primitives to the model. See Figure 2.




grobject
id
region
viewcontext
linkuri
screentip
name
visibility
interactivity
opt
rep
cho
grobject
para
grnode
gdata


para
id
region
viewcontext
linkuri
screentip
name
content
visibility
interactivity
opt
rep
cho
subpara
gdata

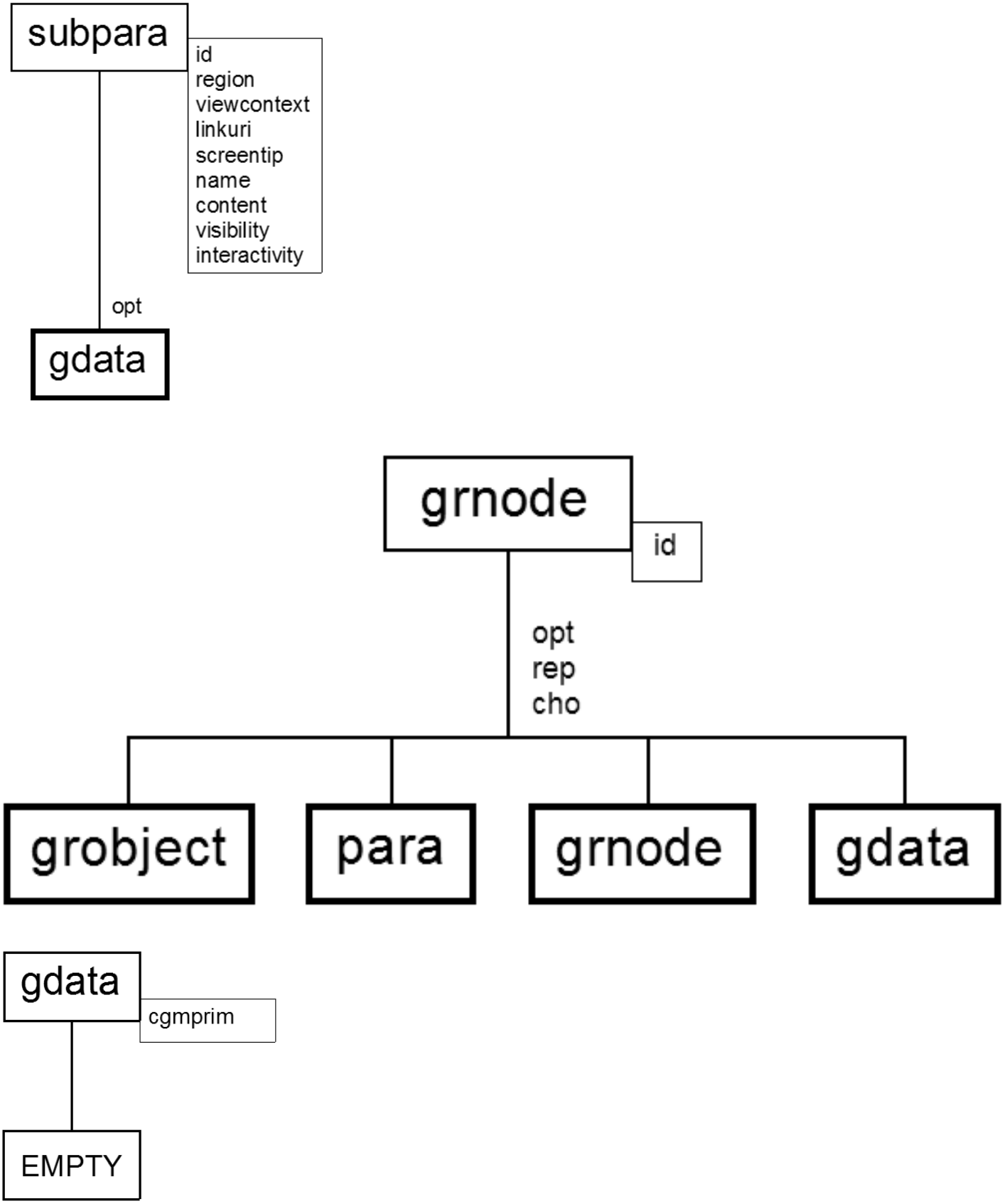


Figure 2. WebCGM Structure Diagrams

2.3.6 Hyperlinking

WebCGM supports bi-directional hyperlinking within individual WebCGM instances, between WebCGM instances and other Web media types.

In-line linking is supported, from WebCGM objects (APS of type 'grobject', 'para', and 'subpara') to WebCGM graphic files, objects, as well as to text and other media types. WebCGM fully supports linking from other media to WebCGM files and objects.

Links from WebCGM objects are realized as 'linkuri' APS Attribute elements contained within the definitions of the objects. The address of the link (the first parameter of the 'linkuri') is any valid URL according to the rules of RFC-2396.

Objects may contain multiple 'linkuri' APS attribute instances, for which case the associated Link Title parameter is available to help the user select the destination. WebCGM prescribes a uniform viewer requirement to offer destination choice to the user for such multi-destination cases.

The target of a link, either from within a WebCGM or from another media type (e.g., HTML text), may be a WebCGM instance. WebCGM defines an optional "fragment syntax" for addressing objects within a WebCGM metafile. A "#" character terminates the base URL, and following the "#" is the WebCGM syntax for object addressing, and for specifying viewer behavior upon execution of the link.

The fragment syntax, in full generality, is:

<base-URL>#<pict-part>.<obj-part>

The <pict-part> is identified by a keyword and has two pieces, the picture locator (either the 'PictureId' string parameter of the CGM, or the picture sequence number), and viewer behavior upon navigating to the picture. With the WebCGM 2.0 restriction of one picture per metafile, the <pict-part> is not useful, but is maintained in the syntax for backward compatibility (with WebCGM 1.0 implementations).

The <obj-part> similarly is identified by a keyword and has two pieces, the Id parameter of the object (APS), and viewer behavior.

The syntax is well-defined so that in many common cases, keywords and pieces can be eliminated and defaulted. So, for example, <base-URL>#<string> unambiguously identifies the object (APS) whose Id parameter is "<string>" in the first (only) picture of the metafile pointed to by "<base-URL>".

See the normative specifications of Chapter 3 for complete details and examples.

2.4 Encodings

ISO CGM defines two encodings of the CGM functionality: Binary, and Clear Text. WebCGM, like other leading industry profiles of CGM, limits the encoding to Binary for the purposes of conforming interchange. It is the Binary encoding which is registered as a MIME type. Using available encoding converters, the Clear Text encoding can be used for debugging, hand authoring, demonstration, etc.

2.5 Graphical content of WebCGM

The graphical content of WebCGM is chosen to balance the requirements of high expressive power, and simplicity to implement. It is a subset of the Model Profile (MP) of the CGM standard. It is closely aligned to the graphical content of industry-specific Web technical profiles (e.g., ATA), and continuous collaborative efforts aim to maintain maximal alignment.

The WebCGM Profile Pro-forma (PPF), later in this document, gives the complete normative graphical content details. Following is a conceptual summary.

2.5.1 Graphical primitives

The most obvious aspect of a graphics format is the collection of graphical primitives - those drawing elements which define the geometric and other presentation content of the format. CGM:1999 contains a rich selection of vector graphics primitives, plus fully integrated state-of-the-art compressed tile raster content.

WebCGM 2.0 includes most of the significant of the graphical drawing primitives of CGM:1999.

- The simple generic drawing primitives - polylines and disjoint polylines, polygons, polygon sets.
- A number of specialized objects, for convenience and efficiency - rectangles, circles and ellipses, circular and elliptical arcs and pie slices. These come in both unfilled (line) and filled flavors.
- Graphical Text primitives:
    - WebCGM allows the simple Restricted Text primitive of CGM:1999 (which carries its extent box with it), as well as the Append Text element (continuation of a text string, after change of attributes such as font, color, ...)
    - WebCGM also supports text-on-path: paths other than simple straight lines can be defined (similar to Compound Line), and the text is laid out along that path.
- Closed Figure and Compound Line - these primitives allow the construction of complex paths as a concatenation of any of the other line and fill primitives. The path may either be drawn according to CGM line attributes, or filled according to any of the permissible fill attributes.
- Smooth curves, available in two flavors:
    - the basic and popular piece-wise cubic Bezier capability of the CGM:1999 Polybezier element;
    - more powerful Non-uniform B-splines (NUBS) and Non-uniform Rational B-splines (NURBS).
- Raster capabilities are available in two flavors, fully integrated with the scalable vector functionality of WebCGM:
    - the simple, uncompressed (except for run-length) Cell Array element of CGM:1999;
    - and the compressed, tiled Tile Array capability of CGM version 3. The popular Web compression formats of CCITT Group 4, PNG, and JPEG are among the supported compression types.

In CGM:1999 but excluded from WebCGM 2.0 are:

- The unrestricted TEXT element is prohibited (the results are unpredictable, so the more reliable RESTRICTED TEXT is required).
- The Hyperbolic Arc and Parabolic Arc elements of CGM:1999 are prohibited (these are seen in some advanced engineering formats, but not yet in Web practice).
- Polysymbol - the Polysymbol element allows the sizing and placement into CGM pictures of "symbols", which are defined in an external Symbol Library (which itself is a CGM).

2.5.2 Attributes and controls

Attribute elements and control elements determine the details of the appearance of graphical primitives.

- Line attributes of WebCGM include line dash styles, line width, line color, and the controls over appearance of line cap and line join. Dash styles can either be predefined, or can be defined precisely by the generator of the metafile.
- Fill attributes control the appearance of the interior of filled primitives. One attribute controls the overall style of the filled-area interior - solid color, hatch, bitmap-style pattern, empty, and hollow. In the case of hatch interior, the style can either be one of a handful of predefined, or can be precisely defined in the metafile. The pattern can be defined as a small "raster tile" of two or more colors.
- The related Edge attributes control the appearance of edges of filled areas, and are for the most part identical to Line attributes.
- The appearance of graphical text can be controlled by the attributes of font and character set (see later), text size, orientation, inter-character spacing, expansion-compression of nominal character aspect rations - the important aspects needed for precise control in modern graphical text presentation. The orientation attribute actually give control not just of rotation, but skewness and aspect distortion, i.e., it is equivalent to a local transformation on text elements.
- WebCGM 2.0 also supports the Generalized Text Path capability (text on arbitrary, curved paths), with the style (Generalized Text Path Mode) limited to 'axis-tangential'.
- WebCGM clipping support includes:
    - rectangular regions;
    - arbitrary complex clip regions (Protection Region, similar to Closed Figure), with Protection Region Indicator restricted to 'clip'.

The following attribute and control features of CGM:1999 are excluded from WebCGM 2.0.

- The use of the bundled attributes model of CGM Version 1 is prohibited, which eliminates all of the Version 1 bundle index elements, the ASF elements, and the bundle-table setting elements of Version 2.
- The use of the CGM Version 2 Segment functionality is prohibited, and its dozen or so associated elements may not occur in valid WebCGM metafiles.
- The Geometric Pattern fill capability is prohibited.
- Shielding functionality is excluded, i.e., Protection Region with mode (Protection Region Indicator) 'shield'.

2.5.3 Colour and transparency

The normal behavior of CGM:1999 viewers is to render later occurring primitives completely opaquely on top of earlier primitives. Two notions of transparency are supported in WebCGM 2.0:

- First, a WebCGM picture can be placed with transparent background via the TRANSPARENT parameter of the OBJECT tag.
- Second, per-element and per-pixel translucency (on a continuum between fully opaque and fully transparent) can be defined via an alpha-transparency Escape element specified in the ISO Register of Graphical Items.

The full range of standard CGM:1999 color models is limited in WebCGM. The default RGB model is included, as well as the models: RGB-alpha; the colorimetric RGB space of the Web, sRGB; and sRGB-alpha. The latter three are registered in the ISO Register of Graphical Items.

2.5.4 Character sets and fonts

Fully international text is supported in WebCGM by:

- Allowing Unicode, UTF-8 or UTF-16, to be selected in the character set designation and selection mechanisms.
- Requiring the use of the CGM:1999 Font Properties element, in the case of fonts outside of a required core set of 13. The Font Properties element carries a handful of parameters which are descriptive of a set of font attributes which are most commonly used to classify and call out fonts.

The default character set is "ISOLatin1". A core set of 13 fonts, the same as those in the ISO CGM Model Profile (MP), are required in WebCGM implementations.

2.6 WebCGM XML Companion File (XCF)

The WebCGM XML Companion File (XCF) is a new significant new component of WebCGM, forecast in 1.0 and added in the 2.0 upgrade. The WebCGM XCF provides a standard way to externalize metadata from a WebCGM instance, while maintaining a tight binding of that metadata to objects (APSs) in the WebCGM instance.

The WebCGM XCF was designed with three main usage scenarios in mind. A WebCGM companion file:

1. can be used as a scaled down XML representation of a WebCGM illustration by enumerating the Application Structures IDs, types and attributes.
2. can be used to bind application specific metadata (such as a part number) to a particular Application Structure in a WebCGM illustration.
3. could also be used to update metadata in a WebCGM illustration via the WebCGM DOM (see DOM section *Relationship withXML companion file* for more information).

Examples and more details may be found in the *WebCGM XML Companion File* (XCF) chapter, the complete normative definition of the WebCGM XCF.

The normative definition of XCF includes a base and generic DTD. The WebCGM XCF is designed to be extensible, by other profiles derived from WebCGM 2.0, as well as applications of WebCGM 2.0. In particular, this allows industry specific metadata to be added to the WebCGM object model. See the normative XCF definition for details.

The XCF is a mechanism to bind external metadata to objects in WebCGM instances. Accordingly, unlike hierarchical tree structured WebCGM instances, the structure of the XML Companion File is mostly flat. See the normative section, *Relationship with XML companion file*, for more details.

2.7 WebCGM Document Object Model (DOM)

2.7.1 Motivation

An interface for programmatic access to WebCGM contents and structure, as well as facilities to manipulate a standardized WebCGM XML Companion File, were perhaps the strongest driving requirements for the upgrade of WebCGM 1.0 to WebCGM 2.0. Virtually all of the WebCGM viewer and user agent implementations had already defined and implemented a proprietary application programming interface (API) for such functionality. Such an interface is part of a Document Object Model (DOM) for WebCGM.

2.7.2 Scope of WebCGM DOM

Compared with detailed, complete DOM specifications such as W3C's XML DOM Level 3, or the DOM of the SVG 1.1 Recommendation, the WebCGM DOM has limited scope. A full DOM would support query and discovery of all objects and entities in a target content (graphic) instance, right down to the leaf nodes of the structure tree. It would also support symmetric, detailed modification and manipulation capabilities for changing the object.

The functionality available in the WebCGM DOM is somewhat more limited. The WebCGM DOM exposes the document graphic structure down to the Application Structure (APS) level -- APS's are the fundamental addressable graphical objects in WebCGM, and are the building blocks of the hierarchical structure tree of a WebCGM.

The WebCGM DOM supports transient manipulation of the APS attributes -- which represent non-graphical metadata associated with the objects -- and transient changes to presentation style of graphical objects. The WebCGM DOM therefore provides the functionality to query and discover the structure of a WebCGM, enumerate its graphical objects, extract associated metadata (e.g., hyperlinking data) from documents, and finally provides users with *standard* ways to add more interactivity to WebCGM documents than was possible in WebCGM 1.0.

The WebCGM DOM also provides functionality for manipulation and application of standard WebCGM XML Companion Files, described in the previous section.

The WebCGM DOM support a number of usage scenarios and gives access to a number of useful capabilities -- see the *WebCGM 2.0 Requirements* for details about the in-scope and out-of-scope capabilities of WebCGM DOM.

Back to top of chapter

---

3 WebCGM Intelligent Content

This section and its subsections are normative, unless otherwise indicated.

3.1 Addressing objects

3.1.1 URI fragment specification

3.1.1.1 Fragment definition

The URI (Uniform Resource Identifier) is how resources are identified on the Internet. For example, a CGM file called web.cgm might have the following URI:

```
http://example.org/web.cgm
```

Objects (application structures) within a WebCGM are addressed using the mechanism of the URI fragment. These WebCGM rules are derived from and are consistent with the Web protocols defined in RFC-2396.

A URI fragment is the separator character " # " appended to the main part of the URI or the "base", followed by a string. The fragment string is usually specific only to a particular class of applications. This clause defines the WebCGM fragment which allows WebCGM viewers, web browsers, scripting engines, and other applications to:

- address (point to) specific objects within WebCGM files,
- or load and apply an XML companion file (XCF).

The base URI is terminated by "#", and a fragment is appended which defines the object, and the desired viewer behavior. The URI fragment syntax is based on concepts described in the *XML Pointer Language* (Xpointer). The URI fragment syntax is defined below. The formal grammar for the WebCGM fragment is given using a simple Extended Backus-Naur Form (EBNF) notation.

Note. Multiple pictures per WebCGM instance were allowed in the previous edition, WebCGM 1.0. WebCGM 2.0 allows only one picture per metafile. For backward compatibility with existing viewer implementations, the fragment syntax is unchanged with regard to the picterm.

3.1.1.2 Fragment EBNF

```
webcgmfragment ::= picterm "." objterm |
                   picterm |
                   objterm  |
                   picid "." objid |
                   objid |
                   xcfterm

picterm ::= pictureid | pictsequence
```

```
pictureid ::= "pictid("   picid   ("," behavior)? ")"

picid ::= (char)+

behavior ::=  "_blank" | "_self" | "_parent" | "_replace" | "_top" | target

target ::= (char)+

pictsequence ::= "pictseqno(" picseqno ("," behavior)? ")"

picseqno ::= "1"

objterm ::= objectid | objectname

objectid ::= "id("   objid   ("," objbehavior)? ")"

objid ::= (char)+

objbehavior ::= "zoom" | "move" | "highlight" |
                "zoom_all" | "move_all" | "highlight_all" |
                "highlight_zoom" | "highlight_move" |
                "highlight_zoom_all" | "highlight_move_all" |
                "view_context"

objectname ::= "name("   objname  ("," objbehavior)? ")"

objname ::= (char)+

xcfterm ::= "xcf(" xcfurl ")"

xcfurl ::= (char)+
```

See next section for a definition of the "char" production.

3.1.1.3 Fragment Character Repertoire

The productions 'picid', 'target', 'objid', and 'objname' in the fragment grammar are represented by parameters in WebCGM of type non-graphical text (CGM type SF). Their character repertoire shall be restricted as follows.

1. Firstly, per the character set for type SF data. See section 6.3, T.14.5.
2. Secondly, the character repertoire for all of these productions is further restricted as defined in section 2.2 of XML, second edition, REC-xml-20001006.
3. Thirdly, the repertoire for each of these productions is further restricted as follows:
    - objid - corresponds to WebCGM APS 'id' parameter; further restrictions: as the *name* construct defined in section 2.3 of REC-xml-20001006.
    - objname - corresponds to WebCGM APS attribute of type 'name'; shall not contain any leading or trailing whitespace (#x09 | #x0a | #x0d | #x20); no further restrictions.
    - picid - corresponds to the 'id' parameter of WebCGM BEGIN PICTURE elements; further restrictions: as objid for any picid value occurrences within a fragment, otherwise per type SF for the id parameter in the BEGIN PICTURE element itself (see further comments below).
    - target - further restrictions: as objid, plus further restricted per "Frame Target Name" of HTML 4.0 (e.g., must begin with a-zA-Z).

Note that these character repertoires allow one or more of the characters ".", ",", "(", and ")". These are significant characters in the syntax of the WebCGM fragment specification. If any of these four significant characters is to appear in a valid id/name string within a fragment instance, then the fragment shall use the unabbreviated long form, which is

the first of the five optional forms in the 'webcgmfragment' production of 3.1.1.2. In particular, all components of the long form shall be included, and none of the parts marked as optional in the EBNF may be omitted.

Note about `picid`: The character repertoire for `picid` occurrence in the fragment ("as `objid`") is more restrictive than the repertoire for the id parameter of the BEGIN PICTURE element itself (CGM data type SF, not further restricted). Any application which intends to use the picid field in the fragment must generate the picture ids to the more restrictive repertoire of the fragment. Note also, since WebCGM 2.0 does not allow multi-picture metafiles, that the `picid` in the fragment is of limited utility. It could perhaps be used as for quality assurance, for verification that the proper picture/metafile is being processed, but it has no picture-selection usage as in multi-picture metafiles.

3.1.1.4 Non-ACSII characters in URIs

The URI character repertoires, as define in the appropriate W3C specifications, basically comprise the alphabetic and numeric characters of ASCII, plus a few punctuations. The character repertoires defined in 3.1.1.3 are potentially much richer. The method for handling this disparity is as defined in section 4.2.2 of XML 1.0, second edition, REC-xml-20001006. That specification is adapted for WebCGM processors as follows.

A WebCGM processor shall interpret a non-ASCII character in a URI by representing the character in UTF-8 as one or more bytes, and then escaping these bytes with the URI escaping mechanism (i.e., by converting each byte to %HH, where HH is the hexadecimal notation of the byte value).

3.1.2 Fragment parameters

3.1.2.1 Picture selection keywords

Note. As noted previously, the picture selection keywords are of limited utility in WebCGM 2.0, which allows only one picture per metafiles. However, they are kept in the syntax for backward compatibility with WebCGM 1.0 (which allowed multi-picture metafiles), and they are a required part of the syntax if there is a `picterm` production present (see EBNF).

`pictid` - The `pictid` keyword indicates that the identity of the picture to be viewed is by the id of the picture, which is the id parameter in the BEGIN PICTURE element. In the syntax, this is a required parameter of the `picterm` production, and there may be a second associated parameter, whose value is an optional picture behavior specification (see EBNF). If the metafile does not contain a picture with the specified picture id value, the first (and only) picture in the metafile is chosen.

`pictseqno` - The `pictseqno` keyword indicates that the identity of the picture to be viewed is by the sequence number of the picture in the metafile. "1" is the only valid value in WebCGM 2.0 (see EBNF). In the syntax, this is a required parameter of the `picterm` production, and there may be a second associated parameter, whose value is an optional picture behavior specification.

3.1.2.2 Picture behaviors

Picture behaviors describe to the viewer how to display a picture that is the target of a hyperlink. Picture behaviors are based on the syntax and semantics of " Frame Target Names" defined in the HTML 4.0 Specification.

The reserved names listed below describe the various picture behaviors. All other Picture Behavior values shall be valid Frame Target Names as described in the HTML 4.0 specification. Frame Target Names must begin with an alphabetic character (a-zA-Z).

In what follows, the following conventions apply:

- "viewer" refers collectively to a browser and/or a CGM viewer (e.g., plugin). The logical result is described, not the method of achieving the result (which in general will require action by browser or viewer or both.)
- "content" refers to either source or destination or a link, and is either a picture (CGM) or a document (HTML), depending upon the formulation of the particular link.

The following Picture Behavior values, except for _replace, are defined per HTML 4.0:

_blank
The viewer shall load the designated content in a new, unnamed window.

_self
The viewer shall load the content in the same frame as the one containing the content that refers to this target.

_parent
The viewer shall load the content into the immediate FRAMESET parent of the current frame in which the current content is displayed. This value is equivalent to "_self" if the current frame has no parent.

_replace
The viewer shall replace the current CGM picture by the designated CGM picture in the same rectangular area in the same frame as the picture which refers to this target. Applicable only to CGM-to-CGM links (i.e., this is not defined in HTML 4.0), this is the default behavior for such links.

_top
The viewer shall load the content into the full, original window (thus canceling all other frames). This value is equivalent to _self if the current frame has no parent.

If the picture behavior value is any valid name string other than the above reserved names, (it begins with an alphabetic character (a-zA-Z)), remove the existing content (picture or document) from the frame whose name matches the string and display the specified content in the specified frame. If no frame exists by the specified name, the viewer shall load the designated content (picture or document) in a new window with the specified name.

In the case that a link is HTML to CGM, the picture behavior should not be included in any fragment specification. Rather, the effect should be achieved with HTML TARGET attribute on the link specification in the HTML, per the specifications in the HTML 4.0 Recommendation. CGM viewers shall ignore picture behavior specifications in URI fragments which are part of links from non-CGM content.

The following table summarizes the rules for other link types, CGM-to-HTML and CGM-to-CGM:

Picture behaviors and different source-to-destination data types.

Behavior	CGM-to-HTML	CGM-to-CGM
_blank	The viewer shall load the designated document in a new, unnamed window.	The viewer shall load the designated picture in a new, unnamed window.
_self	The viewer shall load the document in the same frame as the one containing the CGM picture that refers to this target. This is the default for CGM-to-HTML links.	The viewer shall load the picture in the same frame as the one containing the CGM picture that refers to this target.
_parent	The viewer shall load the document into the immediate FRAMESET parent of the current frame in which the current picture is displayed. This value is equivalent to "_self" if the current frame has no parent.	The viewer shall load the picture into the immediate FRAMESET parent of the current frame in which the current picture is displayed. This value is equivalent to "_self" if the current frame has no parent.

_replace	Not applicable. Default to_self.	The viewer shall replace the current picture by the designated picture in the same rectangular area in the same frame or window as the picture which refers to this target. This is the default behavior for CGM-to-CGM links.
_top	The viewer shall load the document into the full, original window (thus canceling all other frames). This value is equivalent to _self if the current frame has no parent.	The viewer shall load the picture into the full, original window (thus canceling all other frames). This value is equivalent to _self if the current frame has no parent.
target	The viewer shall load the document into the frame identified by "target". If no matching frame can be found, the viewer shall load the designated content document in a new window with the specified name.	The viewer shall load the picture into the frame identified by "target". If no matching frame can be found, the viewer shall load the designated content document in a new window with the specified name.

Note: Link interactions between CGM and non-CGM mime types other than HTML follow the same rules as CGM and HTML interactions.

Figures 3 and 4 below give examples of _self and _replace.

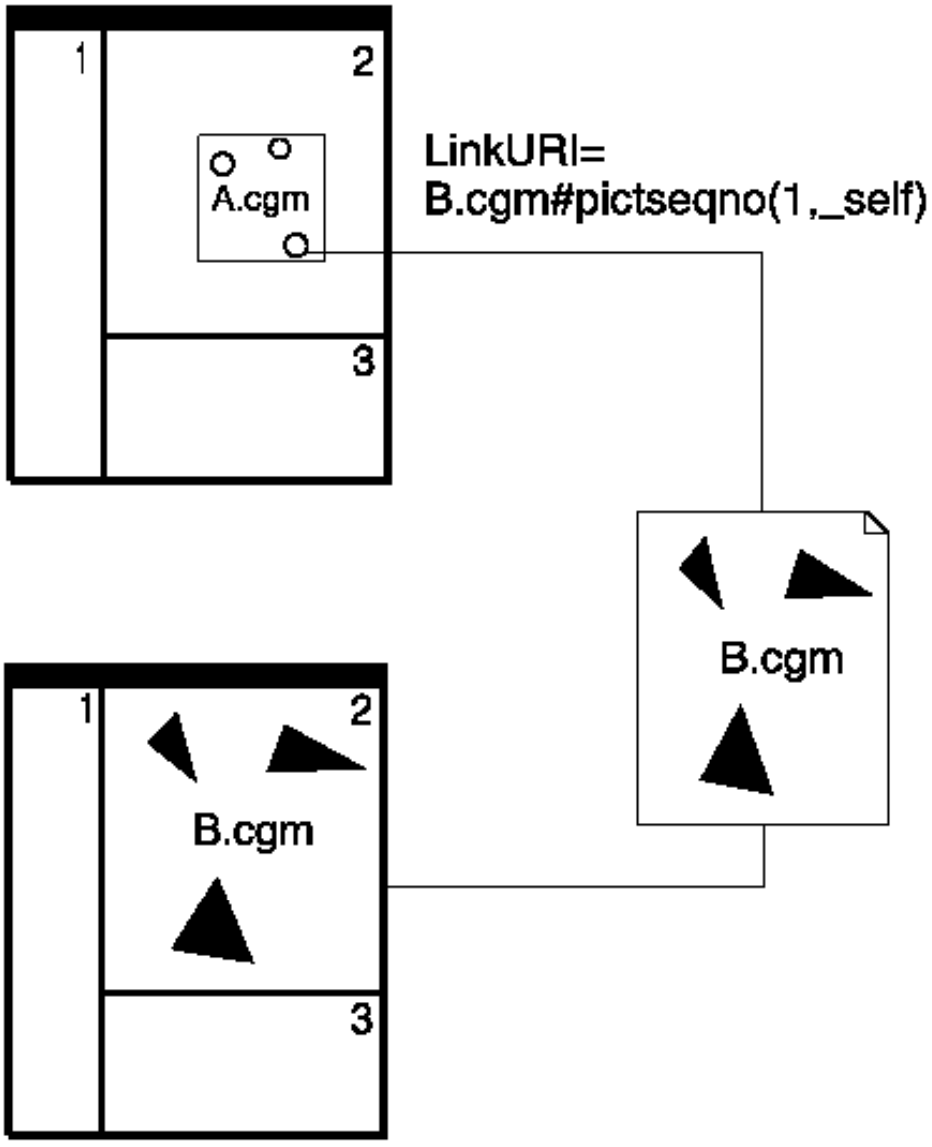


Figure 3. Example of _self

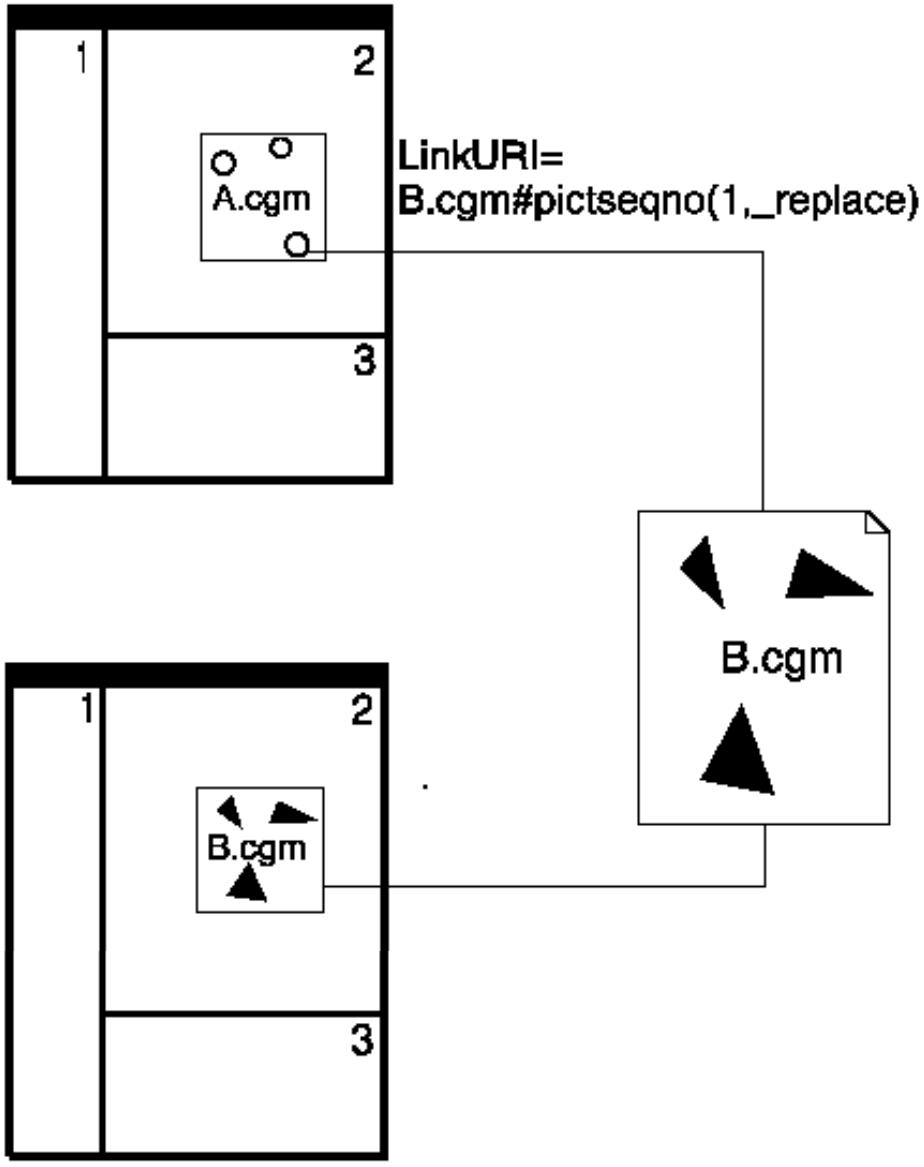


Figure 4. Example of _replace

3.1.2.3 Object selection keywords

`id` - the id of the APS of type 'grobject', 'para' or 'subpara' to be selected. The id parameter in the BEGIN APS element. If no match is found in the picture, no object is selected.

`name` - the value of the 'name' attribute in an object (grobject, para, or subpara APS). This is an alternate way to address objects. The first object in the picture which contains a 'name' attribute with a matching value is selected. If no match is found in the picture, no grobject is selected. If more than one match is found in the picture the object behaviors can be used to select the treatment of all selected objects.

3.1.2.4 Object behaviors

3.1.2.4.1 Enumeration of behaviors

Object behaviors of WebCGM 2.0

Behavior	Navigation	Highlighting	_all
zoom	X		
move	X		
highlight		X	
highlight_all		X	X
zoom_all	X		X
move_all	X		X
highlight_zoom	X	X	
highlight_move	X	X	
highlight_zoom_all	X	X	X
highlight_move_all	X	X	X

Object behavior legal but deprecated in WebCGM 2.0

Behavior	Navigation	Highlighting	_all
view_context	X		

3.1.2.4.2 Definition of target rectangle

If navigation to an object is indicated in the above tables, a target rectangle must be calculated as follows:

- If the object APS contains a 'viewcontext' APS attribute, the target rectangle is described by the 'viewcontext' APS attribute.
- If the object APS does not contain a 'viewcontext' attribute, but contains a 'region' attribute, the target rectangle is the rectangle enclosing the region defined by the 'region' APS attribute.
- If the object APS contains neither a 'viewcontext' nor a 'region' attribute, the target rectangle is the bounding box of the graphical primitives of the object.
- For multiple selected objects the target rectangle is calculated from the bounding box of the target rectangles of the individual objects.

3.1.2.4.3 Definition of behaviors

zoom
: The viewer shall fit the target rectangle of the object into the viewer's rectangle and center it. The "zoom" behavior is the default behavior in WebCGM 2.0.

move
: The viewer shall move (pan) the illustration such that the target rectangle is centered inside the viewer's rectangle. If the target rectangle is too large, the viewer shall fit the target rectangle into the viewer's rectangle and center it.

highlight
: Highlight the first object selected and ignore the 'viewcontext' attribute, if present. The resulting view is a full-picture view, not a zoomed view.

highlight_all
: Highlight all objects selected. The resulting view is a full-picture view, not a zoomed view.

zoom_all
: Same as `zoom`, but using the combined target rectangles of all objects selected.

move_all
: Same as `move`, but using the combined target rectangles of all objects selected.

highlight_zoom
: Same as the combined effects of `zoom` and `highlight`.

highlight_move
: Same as the combined effects of `move` and `highlight`.

highlight_zoom_all
: Same as the combined effects of `zoom_all` and `highlight_all`.

highlight_move_all
: Same as the combined effects of `move_all` and `highlight_all`.

view_context (deprecated)
: If a 'viewcontext' APS attribute exists for the object APS, display only the rectangular region of the picture defined by the 'viewcontext' APS attribute, and highlight the object. If no 'viewcontext' APS attribute exists in the object, the `highlight` behavior shall be implemented.

3.1.2.5 Zoom and pan

In addition to being able to meet the zooming requirements imposed by the object behavior specifications, and those zooming requirements imposed by support of the OBJECT tags paramenters, viewers which operate in an interaction-capable environment shall have zoom and pan controls available to the user. The exact methods and user interface styles for zoom and pan selection and manipulation are viewer dependent.

3.1.2.6 XML Companion File

The fragment syntax can be used to load and apply an XML Companion File (XCF) together with the WebCGM file. The interpreter will load the WebCGM file first. Then it will load and apply the XML Companion File specified by the URI, as defined in RFC2396. A viewer must apply the XCF before first display of the CGM. Interpreters without support for the WebCGM DOM will ignore this fragment type.

3.1.3 Examples

This subsection is informative (non-normative).

3.1.3.1 Preliminaries

The WebCGM fragment in its most verbose form provides the means to address objects between metafiles, and tells the viewer what to do to execute the link. The default viewer behavior defines what the browser shall do if the WebCGM fragment does not explicitly define the viewer behavior.

The following examples illustrate some of the ways the WebCGM fragment can be used. The examples describe how one might address a set of CGM files relating to various views of an engine which are stored on the CGM Open website (`http://example.org/webcgm/`). The CGM files contain various views of an engine assembly - the top view, the front view, the right view, the left view, and the isometric view. The CGM files are identified as follows:

Metafile 1: "engine_top.cgm"

Metafile 2: "engine_front.cgm"

Metafile 3:"engine_right.cgm"

Metafile 4:"engine_left.cgm"

Metafile 5: "engine_iso.cgm"

Each metafile contains several identifiable objects - the oil pump, the cylinder head, the fan, the radiator, or the distributor. Not all objects are shown in all views.

The objects contained in each metafile are as follows:

Metafile 1:

Oil pump: id='oil-pump-t' name='lube-system'

Cylinder head: id='cyl-hd-t' name='engine'

Fan: id='fan-t' name='cooling'

Radiator: id='rad-t' name='cooling'

Distributor: id='dist-t' name='ignition'

Metafile 2:

Oil pump: id='oil-pump-f' name='lube-system'

Cylinder head: id='cyl-hd-f' name='engine'

Fan: id='fan-f' name='cooling'

Radiator: id='rad-f' name='cooling'

Distributor: id='dist-f' name='ignition'

Metafile 3:

Oil pump: id='oil-pump-r' name='lube-system'

Cylinder head: id='cyl-hd-r' name='engine'

Fan: id='fan-r' name='cooling'

Radiator: id='rad-r' name='cooling'

Metafile 4:

Cylinder head: id='cyl-hd-l' name='engine'

Fan: id='fan-l' name='cooling'

Radiator: id='rad-l' name='cooling'

Distributor: id='dist-l' name='ignition'

Metafile 5:

Oil pump: id='oil-pump-i' name='lube'

Cylinder head: id='cyl-hd-i' name='engine'

Fan: id='fan-i' name='cooling'

Radiator: id='rad-i' name='cooling'

Distributor: id='dist-i' name='ignition'

3.1.3.2 Example 1

```
http://example.org/webcgm/engine_top.cgm#pictseqno(1,_blank).id(cyl-hd-t,highlight)
```

When used as the value of the URI attribute in an object in a CGM file, this URI retrieves engine_top.cgm CGM file from the example.org web site and displays the first picture in a new window, highlighting the object with an id of "cyl-hd-t". The existing display remains. The entire picture is displayed, regardless of the presence of a 'viewcontext' APS for the object "cyl-hd-t". Note that inclusion of the picture behavior, "_blank", in the URI fragment is not the preferred formulation for this link/behavior combination (see 3.2.2.3). The preferred form would place the string "_blank" into the 3rd parameter of the 'linkuri' APS attribute, and the first parameter would be:

```
http://example.org/webcgm/engine_top.cgm#pictseqno(1).id(cyl-hd-t,highlight)
```

3.1.3.3 Example 2

```
http://example.org/webcgm/engine_top.cgm#pictseqno(1).id(oil-pump-t,highlight)
```

When used as the URL in an OBJECT element in HTML, this example displays the CGM inside a rectangle defined by the width and height parameters of the OBJECT tag, displaying the whole picture with the pump highlighted.

3.1.3.4 Example 3

```
http://example.org/webcgm/engine_iso.cgm#pictseqno(1,topframe).id(dist-i,view_context)
```

When used as the value of the URI attribute in a object in a CGM file, this URI retrieves the engine_iso.cgm CGM file from the example.org web site and displays the picture in the metafile in the frame named "topframe", highlighting the object with an id of "dist-i". If present the 'viewcontext' APS attribute for the object "dist-i" is used to determine the rectangular portion of the picture to display in the frame. Note, as in example 1, that inclusion of the picture behavior, a target frame named "topframe", in the URI fragment is not the preferred formulation for this link/behavior combination (see 3.2.2.3). The preferred form would place the string "topframe" into the 3rd parameter of the 'linkuri' APS attribute, and the first parameter would be:

```
http://example.org/webcgm/engine_iso.cgm#pictseqno(1).id(dist-i,view_context)
```

3.1.3.5 Example 4

```
http://example.org/engine_front.cgm#picseqno(1,topframe)
```

When used as the value of the URI attribute in a object in a CGM file, this URI retrieves the engine_front.cgm CGM file from the example.org web site and displays the picture in the metafile in the frame named "topframe" Note, as in example 1 and 3, that inclusion of the picture behavior, a target frame named "topframe", in the URI fragment is not the preferred formulation for this link/behavior combination (see 3.2.2.3). The preferred form would place the string "topframe" into the 3rd parameter of the 'linkuri' APS attribute, and the first parameter would be:

```
http://example.org/webcgm/engine_front.cgm#pictseqno(1)
```

3.1.3.6 Example 5

```
http://example.org/webcgm/engine_top.cgm#name(cooling)
```

This example retrieves the engine_top.cgm cgm file from the example.org web site and displays the picture in the metafile. The first object containing a 'name' attribute with value "cooling" is highlighted and the 'viewcontext' APS attribute, if it exists, is used to define the rectangular area of the picture to display.

3.1.3.7 Example 6

```
http://example.org/webcgm/engine_top.cgm#fan-t
```

This example retrieves the engine_top.cgm cgm file from the example.org web site and displays the picture in the metafile. The object with id "fan-t" is highlighted and the 'viewcontext' APS attribute of the object, if it exists, is used to define the rectangular area of the picture to display.

3.1.3.8 Example 7

```
#id(oil-pump-t)
```

This example will highlight the object "oil-pump-t" in the currently displayed picture and the 'viewcontext' APS attribute, if it exists for the object "oil-pump-t", is used to define the rectangular area of the picture to display. The address fragment could be shortened even further to "#oil-pump-t"

3.2 Application Structure and APS Attribute descriptions

3.2.1 Application structures

WebCGM defines these APS types: layer, grobject, para, subpara, and grnode. This document uses the term "object" to refer to an APS of type 'grobject', 'para', and 'subpara'.

Although the picture body of a WebCGM picture is not itself an APS, the content rules of the picture body (`picbody`) are defined by this piece of EBNF:

```
picbody :== layer+ |
            (grobject | para | grnode | gdata)*
```

I.e., a picture body contains either one or more layers, or else it contains a collection of eligible APSs ('grobject', 'para', 'grnode') and graphical data (`'gdata'`). The content rules of these APSs are defined in the following sections.

3.2.1.1 Grobject

Description. The application structure (APS) of type 'grobject' is used to group graphical primitives in a picture together and assign certain attributes to the group. The object is geometrically identified either by the set of primitives enclosed between the BEGIN APS and END APS elements (if any), or by the spatial region associated with the 'region' APS Attribute (if present). 'Grobject' APSs may contain any CGM graphical content allowed by this profile.

Content Model. The permissible content of an APS of type 'grobject' is:

- graphical data
- zero or more APS of type: grobject, para, grnode
- APS attributes of type: region, viewcontext, linkuri, screentip, name, visibility, interactivity.

Viewer Behavior. The selection ("pick") of a 'grobject' APS follows the rules of the WebCGM DOM events [link to event interface]. The 'grobject' has the following specific rules regarding its geometry:

1. If the object contains a 'region' attribute, the region area represents the selectable geometry of the object; otherwise, the drawn primitives represent the selectable geometry of the object. (Note, for filled-area primitives this includes: the edge, if edge visibility is 'on'; the interior, if the interior style is other than 'empty' or 'hollow'; and, the boundary, for interior style 'hollow'.)

It is possible that a user selection ("pick") of a object in a displayed WebCGM picture may not have a single unambiguous result - the cursor location may reside within the geometric extent or 'region' of more than one APS. In such a case, the pick priority is as follows:

2. If the pick has selected only one APS (and it contains a 'linkuri'), then it is the one which is picked;
3. If the pick has selected more than one APS (and each contains a 'linkuri'), and the APSs are not nested, then the APS with the highest pick priority is the one whose picking region occurs latest in the picture (see the 'region' APS Attribute for definition of "picking region").
4. If the pick has selected more than one APS (and each contains a 'linkuri'), and the APSs are nested, the user shall be given a choice of which URI to navigate.

Viewers shall give visual feedback to the user that a successful pick has occurred, and an indication of the particular object (or region) which has been picked. The exact method of feedback is viewer dependent.

If an APS is the target of a link, either from within the picture or from content external to the picture, then the behavior of the viewer shall be as defined in the section "Object behaviors".

The CGM:1999 standard allows the definition of an APS to be continued in pieces which are disjoint in the file. If an APS occurs which has the same value of the 'id' parameter as an earlier APS occurrance, then that is construed as a continuation of the definition of that object. WebCGM 2.0 metafile instances shall not contain continued APS constructs.

3.2.1.2 Layer

Description. The 'layer' APS declares that the graphical content within this APS and any valid nested APS ('grobject' and 'para', but not 'layer') belong to the layer identified by the contained 'layername' APS attribute.

Content Model. The permissible content of an APS of type 'layer' is:

- graphical data
- zero or more APS of type: grobject, para, grnode
- APS attributes of type: layername, layerdesc, visibility, interactivity.

Each 'layer' APS shall contain exactly one 'layername' APS attribute.

Viewer Behavior. Viewers shall provide functionality to inform users of the presence of layers, their names and descriptions. Viewers shall provide functionality to selectively turn on and off the visibility of layers. Viewer may, but are not required to, provide additional functionality for the view manipulation and browsing of layers.

3.2.1.3 Para

Description. The application structure (APS) of type 'para', may be used to identify text ("paragraphs"). In the case that the underlying graphic does not represent graphical text in a searchable form (e.g., the text has been rasterized, polygonized, broken into small units, or pieces of a single string occur in awkward order), 'para' together with 'content' can potentially enable text search functionality. 'Para' APSs may contain any CGM graphical content allowed by this profile.

Content Model. The permissible content of an APS of type 'para' is:

- graphical data
- zero or more APS of type: subpara
- APS attributes of type: region, viewcontext, linkuri, screentip, name, content, visibility, interactivity.

Viewer Behavior. The WebCGM prescription for priority of text search matching is: 'para' with matching 'content' (1st priority match); 'para' without 'content' but with recognizable single-element RESTRICTED TEXT match (2nd priority match); or, single-element RESTRICTED TEXT match, outside of any 'para' (3rd priority match). Further details of behavior of viewers with respect to identifying matches or communicating the identity of matches is not specified in WebCGM.

In other respects, e.g., picking and link navigation, the viewer behavior of 'para' is identical to that of 'grobject'. See 3.2.1.1.

3.2.1.4 Subpara

Description. The application structure (APS) of type 'subpara', may be used to identify smaller fragments of text within APS of type 'para'. This enables, for example, the identification of the larger text block (the "paragraph") for searching purposes, and the tagging of smaller fragments as hotspots. 'Subpara' APSs may contain any CGM graphical content

allowed by this profile, but may not contain nested APS. The APS attribute content rules of sub-para matches those of 'para'.

Content Model. The permissible content of an APS of type 'subpara' is:

- graphical data
- APS attributes of type: region, viewcontext, linkuri, screentip, name, content, visibility, interactivity.

Viewer Behavior. See 3.2.1.3, 'para'.

3.2.1.5 Grnode

Description. The application structure 'grnode' is meant to group basic graphical primitives only. For this reason, 'grnode' must contain the APS 'id' parameter required by CGM:1999 for all APS, but a 'grnode' cannot contain any APS attribute elements. The content of a 'grnode' is however, not limited to graphical primitives. The allowed APS content of a 'grnode' is the same as for the 'grobject' Application Structure.

Like the other Application Structure types, 'grnode' supports inheritance (i.e., it is possible to make a 'grnode' non-visible by inserting it within an object which support the 'visibility' attribute).

Content Model. The permissible content of an APS of type 'grnode' is:

- graphical data
- zero or more APS of type: grobject, para, grnode

Viewer Behavior. Unlike other application structures, 'grnode' is not interactive. It does not receive mouse events like other application structures. The content of a 'grnode' can however be interactive if that content is a direct child of a 'grobject', 'para' or 'subpara'. Additionally, if a mouse event is triggered on the geometry of a 'grnode', an ancestor node may respond to the event if that ancestor node is of type 'grobject', 'para' and 'subpara'. See the Event interface for more information regarding mouse events.

3.2.1.6 About general metadata elements

Description. This version and level of WebCGM do not allow additional APS elements to occur, other than 'grobject', 'layer', 'para', 'subpara', and 'grnode'. Private metadata may be associated with WebCGM objects by keeping the metadata outside of the CGM, and associating it with objects within the CGM. A means for binding external private metadata to WebCGM instances is defined in the section *XML Companion File* (XCF).

3.2.2 Application Structure Attributes

3.2.2.1 Region

Initial value: none
Applies to: 'grobject', 'para', 'subpara'
Inherited: no

Description. The 'region' APS Attribute provides an optional spatial region, associated with a graphical object, which assumes precedence for user picking operations directed at the object. Simple regions of type rectangle, ellipse, polygon, and continuous polybezier can be defined. Complex regions which comprise a collection of simple regions can be built, allowing definition of disjoint subregions, regions with holes, etc. Their semantics (subregions, and interior/exterior definition) are identical to those of the CGM element CLOSED FIGURE. At most one 'region' attribute may be present within a single APS.

Parameters. The data record is a SDR of one or more member pairs (i.e., 2*m members, m>=1). Each member-pair defines a simple region: the first member is of data type Index, whose valid values are:

1. rectangle
2. ellipse
3. polygon
4. continuous polybezier

The second member is type VDC and contains:

- for rectangle: 4 VDC defining two corner points;
- for ellipse: 6 VDC defining respectively the center, and two CDP endpoints;
- for polygon: 2n VDC defining polygon vertex points
- for polybezier: 2*(3n+1) VDC values, representing 3n+1 points, defining n contiguous cubic bezier segments;

For polygon and polybezier regions, closure is implicit (if the last given point does not match the first, then the viewer closes the region with a straight line segment from the last to the first).

In the case that there are multiple simple regions, m>1, then the individual simple regions each correspond to a REGION in the sense of the CGM CLOSED FIGURE element.

Viewer Behavior. See 3.2.1.1.

3.2.2.2 Viewcontext

Initial value: none
Applies to: 'grobject', 'para', 'subpara'
Inherited: no

Description. The 'viewcontext' APS Attribute provides specification to viewers of the initial view of an object, when the viewer has been directed to navigate to the graphical object which contains this attribute. A 'viewcontext' APS Attribute may be contained within an otherwise empty APS, in which case the APS provides only a viewport specification. At most one 'viewcontext' attribute may be present within a single APS.

Parameters. The data record is a SDR of 1 member of type VDC defining two corner points of a rectangle.

Viewer Behavior. See 3.2.1.1.

3.2.2.3 Linkuri

Initial value: none
Applies to: 'grobject', 'para', 'subpara'
Inherited: no

Description. The 'linkuri' APS Attribute defines a URI, to be associated with the object containing this attribute. When the object is selected by a graphical pick operation, then the viewer shall take necessary action to navigate the link. Multiple 'linkuri' attributes may be contained within a single APS. If the object contains more than one 'linkuri' attribute, the user shall be given a choice of which URI to navigate.

Parameters. The data record is a SDR of one member, containing three strings (type SF, String Fixed). The first string is the link destination, a URI, the second string (possibly null) is a Link Title parameter, and the third string is (possibly null) the Behavior parameter. Note that a null string is a zero-length string, and is not the same as an omitted

parameter.

The destination of a link is specified by a Uniform Resource Identifier, or URI. Any valid URL under the specifications of RFC-2396 is a valid value of this parameter. This specification does not constrain the syntax or semantics of a URI in a 'linkuri' that identifies a resource that is not a CGM file (for example, an HTML or XML document).

The Behavior string defines picture behavior associated with the link. The values and meanings are as defined in 3.1.2.2. In cases that the destination is not CGM media type, the 3rd parameter, Behavior, shall be used if picture behavior is to be specified for the link (there is no other option). The Behavior string may also be used for links to CGM media types, and is the preferred method.

In the case that the URI points to CGM media type, the picture behavior may be encoded within the optional fragment identifier in conjunction with the URI structure, per section 3.1.1, "Addressing Pictures and Objects". This form is deprecated, and may be removed in a future edition of this profile. For specifying picture behavior, particular WebCGM 'linkuri' instances shall use either the Behavior string (preferred), or the picture behavior specification embedded in the fragment, but not both.

3.2.2.4 Layername

Initial value: none
Applies to: 'layer'
Inherited: no

Description. The 'layername' APS Attribute declares that the graphics associated with the 'layer' APS containing this attribute belong to the identified layer. The 'layername' need not be unique. If more than one 'layer' APS contains the same 'layername', then the occurrences following the first occurrence shall be construed as continuing the definition of the named layer. Exactly one 'layername' attribute must be present within each 'layer' APS.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed) - the Layer Name (identifier). The string can be null (zero-length). If the Layer Name is null, then the graphics of this object belong to the null layer.

Viewer Behavior. See 3.2.1.2.

3.2.2.5 Layerdesc

Initial value: none
Applies to: 'layer'
Inherited: no

Description. The 'layerdesc' APS Attribute provides optional descriptive text which is associated with the 'layer' APS in which it occurs. This may be used by viewers to facilitate required and optional layer manipulation functions, as described in 3.2.1.2. At most one 'layerdesc' attribute may be contained within a single 'layer' APS.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed).

Viewer Behavior. See 3.2.1.2.

3.2.2.6 Screentip

Initial value: none
Applies to: 'grobject', 'para', 'subpara'

Inherited: no

Description. The 'screentip' APS Attribute provides an optional string, to be associated with a graphical object, which viewers can display when the graphical cursor passes over the graphical object. This APS Attribute may occur within any graphical object of WebCGM, specifically, within any APS of type 'grobject', 'para', and 'subpara', and there shall be at most one occurrence within any particular APS.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed).

Viewer Behavior. Viewers shall be capable of displaying the screen tip, if one is defined for a graphical object, visible to the user when the cursor passes over the graphical object, in the common style of Web browsers.

3.2.2.7 Name

Initial value: none
Applies to: 'grobject', 'para', 'subpara'
Inherited: no

Description. The 'name' APS Attribute provides an optional string, that defines a "common name" associated with an object. Unlike the APS 'id' parameter, the 'name' APS attribute need not be unique within a metafile. Multiple 'name' attributes may be contained within a single APS.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed).

Viewer Behavior. The 'name' gives applications a way to associate common names with objects. The object can optionally be addressed by the value of the 'name' attribute. See 3.1.1.

3.2.2.8 Content

Initial value: none
Applies to: 'para', 'subpara'
Inherited: no

Description. The 'content' APS Attribute provides a means to declare what is the first priority searchable text content of a 'para' APS. The 'content' APS Attribute may occur only within APS of type 'para' and 'sub-para', and there shall be at most one occurrence within any such APS.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed).

Viewer Behavior. See the description under the 'para' APS, 3.2.1.3.

3.2.2.9 Visibility

Initial value: on
Applies to: 'layer', 'grobject', 'para', 'subpara'
Inherited: yes

Description. The 'visibility' attribute indicates if an object is visible or not. 'Visibility' applies to Application Structures (APS) of type 'layer', 'grobject', 'para' and 'subpara'. The value of 'visibility' is inherited by any decendant objects of type 'layer', 'grobject', 'para' and 'subpara' in the WebCGM structure.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed). The valid values

are 'off', 'on'.

Viewer behavior. A non-visible object is not displayed. A non-visible object behaves like a non-interactive object (i.e., it cannot be clicked or highlighted). This does not imply that the 'interactivity' attribute is changed to off, but simply that the user agent must not respond to mouse events.

3.2.2.10 Interactivity

Initial value: on
Applies to: 'layer', 'grobject', 'para', 'subpara'
Inherited: yes

Description. The 'interactivity' attribute indicates if an object may receive mouse events. 'Interactivity' applies to Application Structures (APS) of type 'layer', 'grobject', 'para' and 'subpara'. The value of 'interactivity' is inherited by any decendant objects of type 'layer', 'grobject', 'para' and 'subpara' in the WebCGM structure.

Parameters. The data record is a SDR of one member, containing one string (type SF, String Fixed). The valid values are 'off', 'on'.

Viewer behavior. When the 'interactivity' of an object is set to off, events for this object are disabled. This has the effect of disabling event handlers, cursor changes, highlighting, screentip and hyperlinking for the given node and its descendant. An object that is the target of a link always responds to highlighting, regardless of its 'interactivity' attribute value.

3.2.2.11 About general metadata elements

Description. This version and level of WebCGM do not allow additional APS Attribute elements to occur, other than as enumerated above. Private metadata may be associated with WebCGM objects by keeping the metadata outside of the CGM, and associating it with objects within the CGM. A means for binding external private metadata to WebCGM instances is defined in the section *XML Companion File* (XCF).

3.3 Content model

This subsection is informative (non-normative).

This is an informative, at-a-glance summary of the whole content model of the CGM Version 4 functionality of WebCGM - the "Intelligence" content - using the formal specification technique of the XML DTD. It has been suggested that validating XML parsers could be adapted to perform content validation of WebCGM instances (either via modification of the readers, or via transformation of the intelligent content of WebCGM instance).

```
<!-- To document the structure of the CGM Version 4      -->
<!-- content of WebCGM 2.0 the following DTD fragment    -->
<!-- has been developed.                                 -->
<!--                                                     -->
<!-- PICBODY is included in this DTD fragment for        -->
<!-- purposes of demonstrating that the layer, grobject, -->
<!-- and para structures can exist within the picture    -->
<!-- body level in a CGM instance. The gdata element     -->
<!-- with its associated cgmprim entity attribute is     -->
<!-- intended to represent the model for CGM data stored -->
<!-- as an external entity.                              -->
<!--                                                     -->
<!-- Note: of the attributes listed below, all           -->
```

```
<!-- correspond to APS Attribute elements of CGM, except -->
<!-- the 'ID', which corresponds to the 'id' parameter   -->
<!-- of the CGM APS element.                             -->


<!ELEMENT picbody (layer+ | (grobject | para | grnode
                             | gdata)*)                    >

<!ELEMENT layer (grobject | para | grnode | gdata)+        >
<!ATTLIST layer
  id            ID         #REQUIRED
  layername     CDATA      #REQUIRED
  layerdesc     CDATA      #IMPLIED
  visibility    (on | off) #IMPLIED
  interactivity (on | off) #IMPLIED
                                                           >
<!ELEMENT grobject (grobject | para | grnode | gdata)*     >

<!ATTLIST grobject
  id            ID         #REQUIRED
  region        CDATA      #IMPLIED
  viewcontext   CDATA      #IMPLIED
  linkuri       CDATA      #IMPLIED
  screentip     CDATA      #IMPLIED
  name          CDATA      #IMPLIED
  visibility    (on | off) #IMPLIED
  interactivity (on | off) #IMPLIED
                                                           >
<!ELEMENT para (subpara | gdata)*                          >
<!ATTLIST para
  id            ID         #REQUIRED
  region        CDATA      #IMPLIED
  viewcontext   CDATA      #IMPLIED
  linkuri       CDATA      #IMPLIED
  screentip     CDATA      #IMPLIED
  name          CDATA      #IMPLIED
  content       CDATA      #IMPLIED
  visibility    (on | off) #IMPLIED
  interactivity (on | off) #IMPLIED
                                                           >
<!ELEMENT subpara (gdata)*                                 >
<!ATTLIST subpara
  id            ID         #REQUIRED
  region        CDATA      #IMPLIED
  viewcontext   CDATA      #IMPLIED
  linkuri       CDATA      #IMPLIED
  screentip     CDATA      #IMPLIED
  name          CDATA      #IMPLIED
  content       CDATA      #IMPLIED
  visibility    (on | off) #IMPLIED
  interactivity (on | off) #IMPLIED
                                                           >
<!ELEMENT grnode (grobject | para | grnode | gdata)*       >
<!ATTLIST grnode
```

```
  id            ID         #REQUIRED
                                                          >
<!ELEMENT gdata EMPTY                                    >
<!ATTLIST gdata
  cgmprim       ENTITY     #REQUIRED                       >
```

Note: the use of XML to express the content model of WebCGM implies that a particular attribute can have at most one instance within a particular APS instance. This is not the case, and the normative rules are as specified in 3.2.2.1 through 3.2.2.8.

3.4 WebCGM and the OBJECT tag

The only standard way to reference in-line CGMs from HTML documents is through the OBJECT tag, using the DATA attribute for the CGM file and the TYPE attribute to specify the full Mime Type. The minimal tag for adding CGM into a document would be:

```
<OBJECT DATA="xxx.cgm" TYPE="image/cgm;Version=4;ProfileId=WebCGM"; WIDTH=200;
HEIGHT=100>
```

Other HTML 4.0 attributes which may be used in the OBJECT tag include ALIGN, BORDER, HSPACE, ID, NAME and VSPACE. Use of ALIGN, BORDER, HSPACE, and VSPACE is only permitted in Transitional HTML 4.0, not Strict HTML 4.0.

The attributes CLASSID, CODEBASE, DECLARE, SHAPES, USEMAP, CODETYPE, and ARCHIVE are prohibited. Content with uses WebCGM shall not make direct reference to the code which may be used to display it.

The event-related attributes, ONCLICK,...,ONMOUSEOVER, are permitted but their effect is undefined in this version of WebCGM. Mouse-related events occurring within the area of the WebCGM picture will be handled by the WebCGM viewer, which need not expose these events.

The attributes ACCESSKEY, ALT, CLASS, DIR, LANG, LONGDESC, STANDBY, STYLE, TABINDEX and TITLE are permitted, but have no defined effect on CGM viewers and display of CGM pictures. They are used to improve accessibility, and may also affect the presentation of any alternative text content of the OBJECT element.

The OBJECT element can contain optional PARAM elements, which allow the HTML to pass additional data to the OBJECT. The following PARAMs are defined and permitted for WebCGM. Each PARAM is presented as a name, and permissible values.

FIXED No|Yes
: States whether the CGM is fixed or can be looked at in detail. This is useful to divide icons, logos etc from browseable diagrams. Default is No. Note: for the value Yes, viewers shall not put up scroll bars and zoom options, which would normally be offered for scalable, zoomable pictures.

BACKGROUND Enable|Disable
: States whether the CGM picture shall be drawn with its normal background color, as given in the CGM picture (Enable), or whether the background color of the picture shall be suppressed (Disable), thus allowing the background color or background image of the HTML page to show through. Default is Enable. Note: the background Color of the picture is either default, or is explicitly defined by the BACKGROUND COLOUR element in the CGM picture.

VIEWPORT topx topy botx boty
: Gives a viewport of the CGM (a part of the picture) to display. The values are the top-left and bottom-right corners of the sub-picture. The units are either in the Virtual Device Coordinates (VDC) of the CGM; or, as a percentage of the picture's VDC EXTENT element (whether explicit or default), if the value is followed by a percent sign (%). This facility allows for different parts of a CGM picture to be displayed at different scales in

different places in the document. Default is full VDC EXTENT. Note: the use of VIEWPORT can conflict with some options (e.g., those object behaviors which effectively select a sub-picture via a 'viewcontext' APS attribute within the CGM picture) in a URI fragment on the DATA attribute. In case of conflict, the VIEWPORT specification shall have precedence.

MAPPING fit|fill
HALIGN top|middle|bottom
VALIGN left|middle|right

Each CGM picture has a fixed aspect ratio, determined by the VDC EXTENT element, which may not agree with the aspect ratio defined by HEIGHT and WIDTH specified by the OBJECT tag. These three parameters can be used to specify where and how to place the CGM within the window specified by the OBJECT tag. 'Fit' specifies to isotropically scale the picture (or sub-picture) so that one dimension fits exactly in the window - there will be undrawn space left in the window in one dimension if the aspect ratios don't match. 'Fill' means to isotropically scale so that the window is filled in both directions - if the aspect ratios don't match, a part of the picture will be clipped away at the window boundary.

The default values are 'fit', 'middle', 'middle'.

These PARAMs differ from the ALIGN attribute of OBJECT, which is used to specify where the OBJECT is placed in the document. This could be expressed using the PARAM tag as:

```
<PARAM name="halign" value="middle">
<PARAM name="valign" value="middle">
```

Figures 5 and 6 show the different effects achieved by 'Fit' and 'Fill' with the different alignments.

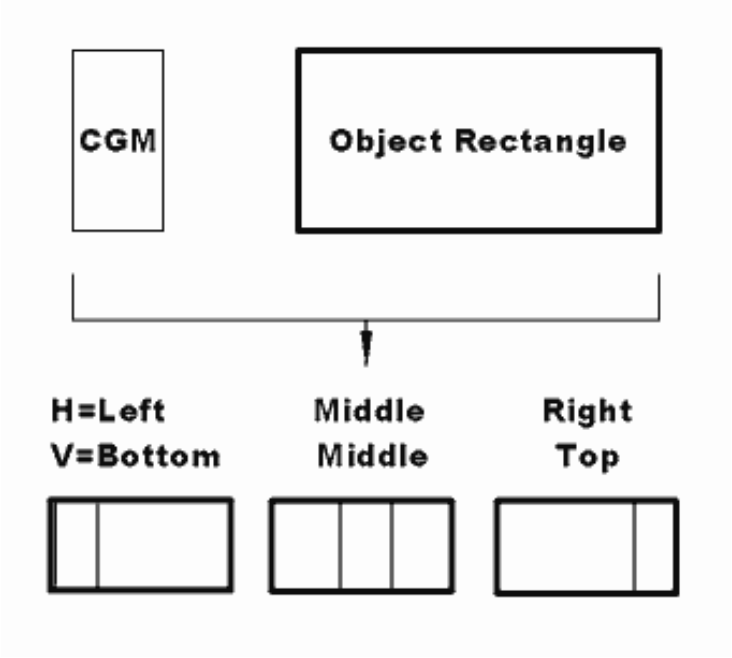


Figure 5. Example of Fit

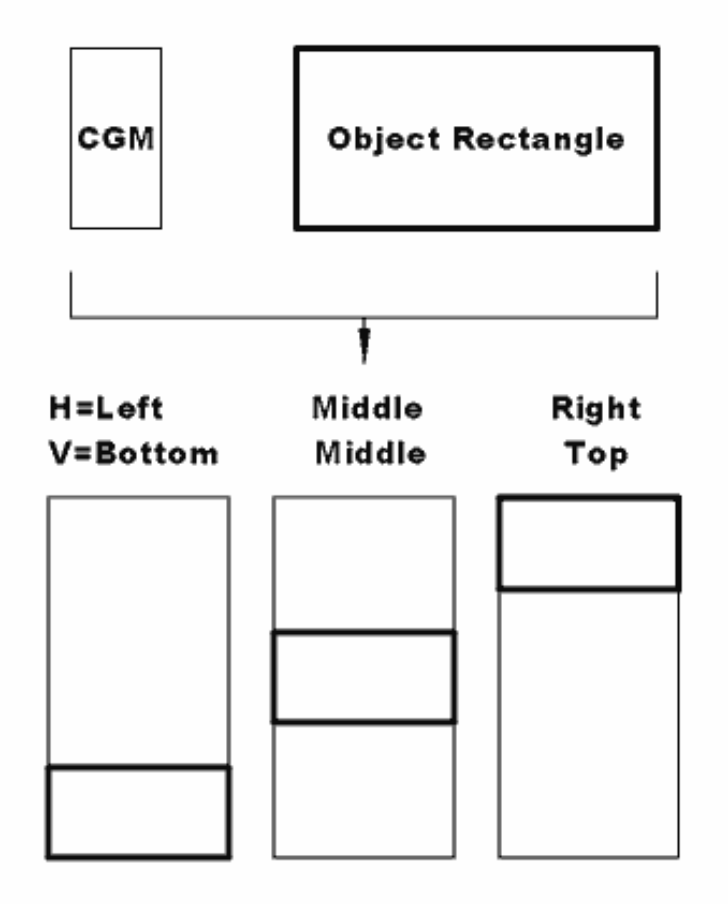


Figure 6. Example of Fill

Back to top of chapter

---

4. WebCGM XML Companion File

This section and its subsections are normative.

Contents

The WebCGM XML Companion file is a new component of WebCGM, forecast in 1.0 and added in the 2.0 upgrade. The element and attribute definitions found in this section represent the WebCGM DTD. As the wording will demonstrate, the DTD may be extended by profiles deriving from WebCGM 2.0. The WebCGM XML companion files may be used for several purposes. There are many conceivable usage scenarios, but for the scope of WebCGM 2.0, the following three were identified as more important. An XML companion file can be used as a scaled down XML representation of a CGM illustration by enumerating the Application Structures IDs, types and attributes.

Example 4.1: *A companion file used as a description of sample_1.cgm. In this case, the description puts an emphasis on the hotsport region of each 'grobject' element*.

```
<webcgm id="root-cgm" filename="sample_1.cgm">
<grobject apsid="id_1" region="1,0,0,100,100"/>
<grobject apsid="id_2" region="1,200,0,300,100"/>
...
<grobject apsid="id_49" region="1,1600,600,1700,700"/>
<grobject apsid="id_50" region="1,1800,600,1900,700"/>
</webcgm>
```

A companion file can be used to bind application specific details (such as a part number) to a particular Application Structure. It is up to the application to control how the application-specific information is used.

Example 4.2: *A companion file used to relate some application-specific data to graphical objects*.

```
<!ENTITY % grobjectAttEXT "model:partNum CDATA #IMPLIED" >
<webcgm id="root-cgm" filename="sample_2.cgm" xmlns:model="http://example.org">
<grobject apsid="id_1" model:partNum="bolt-100A"/>
<grobject apsid="id_2" model:partNum="wingnut-T9"/>
...
<grobject apsid="id_49" model:partNum="drill-bit-D01"/>
<grobject apsid="id_50" model:partNum="router-bit-B389"/>
</webcgm>
```

Or alternatively, a companion file could also be used to update a CGM illustration via the WebCGM DOM (see section *Relationship with XML companion file* for more information):

Example 4.3: *A companion file used to update sample_3.cgm before being displayed by a user agent. Calls to some WebCGM DOM methods need to take place to perform this task*.

```
<webcgm id="root-cgm" filename="sample_3.cgm">
<bindByName apstargetname="bolt_100" screentip="Replacement part: bolt-100B"/>
<bindByName apstargetname="wingnut_9" screentip="Replacement part: wingnut-T9A"/>
</webcgm>
```

Regardless of the usage scenario, a base and generic DTD is required.

4.1 Extending the XML Companion File

The WebCGM DTD is extensible so that industry specific metadata may be added to the WebCGM object model (as shown in example 4.2). The extension definitions are implemented using namespaces. The DTD defines an extension entity for the content and attributes of most elements. As an example, a part manufacturer may want to associate parts information to graphical objects. This might be implemented with an extension that looks like:

```
<!ENTITY   % grobjectAttEXT   "model:partNum CDATA #IMPLIED" >
```

A host application could query the WebCGM DOM and retrieve the associated part information.

A set of rules must be followed when extending the WebCGM DTD:

1. The root element MUST be 'webcgm'.
2. The extended elements and/or attributes MUST be in another namespace.
3. Extending the list of children of an element MUST use the *elementName*EXT entity.
4. Extending the attribute list of an element MUST use the *elementName*AttEXT entity.
5. Elements from the WebCGM namespace MUST NOT be direct or indirect children of other namespaced elements.

The rules found above allow WebCGM user agents to process extended companion files in an interoperable maner.

The structure of the XML companion file is mostly flat. Since it is not the intent of the WebCGM companion file to be a detailed XML representation of a CGM document, there is no need to force the hierarchy of the CGM file to be expressed in the XML companion file. In fact, the WebCGM 2.0 DTD prevents users from doing so. However, it is mandatory to place metadata elements in a companion file according to the hierarchical structure of the CGM file. Without a specific parent node for each metadata element, it would not be possible for WebCGM user agents to know where, in the object model, to insert a metadata. See section *Relationship with XML companion file* for more details.

4.2 The 'webcgm' element

A WebCGM companion file (or any other CGM profile derived from the WebCGM profile) must have a 'webcgm' element as

the root element.

```
    <!ENTITY % webcgmEXT "" >
    <!ENTITY % webcgmAttEXT "" >
    <!ELEMENT webcgm ( (layer | grobject | para | subpara | bindById | bindByName
    %webcgmEXT;)* ) >
    <!ATTLIST webcgm  id       ID    #IMPLIED
                      version  '2.0' #FIXED
                      filename CDATA #IMPLIED
                      %webcgmAttEXT;
    >
```

Attribute definitions:

id="xml:id"
Standard XML attribute for assigning a unique name to an element. Refer to the "Extensible Markup Language (XML) 1.0" Recommendation [XML10].
version="CDATA"
Represents the version of the WebCGM specification. The value is set to 2.0 for this specification. Note: industry specific profiles should use a namespace attribute to identify its profile version.
filename="CDATA"
Represents the filename of the corresponding WebCGM file. 'filename' is a descriptive attribute.
webcgmEXT=""
webcgmEXT is a mechanism for adding additional child content (i.e., metadata) on the root node.
webcgmAttEXT=""
webcgmAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the root node.

4.3 The 'layer' element

The 'layer' element of an XML companion file represents a CGM Application Structure of type 'layer'. The corresponding 'layer' is identifiable given its assigned 'apsid' attribute value.

```
    <!ENTITY % layerEXT "" >
    <!ENTITY % layerAttEXT "" >
    <!ELEMENT layer (%layerEXT;)* >
    <!ATTLIST layer apsid         ID           #REQUIRED
                    layerdesc     CDATA        #IMPLIED
                    visibility    ( on | off ) #IMPLIED
                    interactivity ( on | off ) #IMPLIED
                    %layerAttEXT;
    >
```

Attribute definitions:
apsid="xml:id"
The unique identifier of the Application Structure for the given WebCGM file.
layerdesc="CDATA"
Value of the 'layerdesc' Application Structure attribute for the associated APS.
visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.
interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.
layerEXT=""

layerEXT is a mechanism for adding additional child content (i.e., metadata) on the 'layer'.
layerAttEXT=""
layerAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the 'layer'.

See also the 'layer' functional description in Section 3.

4.4 The 'grobject' element

The 'grobject' element of an XML companion file represents a CGM Application Structure of type 'grobject'. The corresponding grobject' is identifiable given its assigned 'apsid' attribute value.

```
    <!ENTITY % grobjectEXT "" >
    <!ENTITY % grobjectAttEXT "" >
    <!ELEMENT grobject ( linkuri %grobjectEXT; )* >
    <!ATTLIST grobject  apsid         ID           #REQUIRED
                        screentip     CDATA        #IMPLIED
                        region        CDATA        #IMPLIED
                        viewcontext   CDATA        #IMPLIED
                        visibility    ( on | off ) #IMPLIED
                        interactivity ( on | off ) #IMPLIED
                        %grobjectAttEXT;
    >
```

Attribute definitions:
apsid="xml:id"
The unique identifier of the Application Structure for the given WebCGM file.
screentip="CDATA"
Value of the 'screentip' Application Structure attribute for the associated APS.
region="CDATA"
Value of the 'region' Application Structure attribute for the associated APS.
viewcontext="CDATA"
Value of the 'viewcontext' Application Structure attribute for the associated APS.
visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.
interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.
grobjectEXT=""
grobjectEXT is a mechanism for adding additional child content (i.e., metadata) on the 'grobject'.
grobjectAttEXT=""
grobjectAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the 'grobject'.

See also the 'grobject' functional description in Section 3.

4.5 The 'para' element

The 'para' element of an XML companion file represents a CGM Application Structure of type 'para'. The corresponding 'para' is identifiable given its assigned 'apsid' attribute value.

```
    <!ENTITY % paraEXT "" >
    <!ENTITY % paraAttEXT "" >
    <!ELEMENT para ( linkuri %paraEXT; )* >
    <!ATTLIST para apsid         ID           #REQUIRED
                   screentip     CDATA        #IMPLIED
                   region        CDATA        #IMPLIED
                   viewcontext   CDATA        #IMPLIED
                   visibility    ( on | off ) #IMPLIED
                   interactivity ( on | off ) #IMPLIED
                   %paraAttEXT;
    >
```

Attribute definitions:
apsid="xml:id"
The unique identifier of the Application Structure for the given WebCGM file.
screentip="CDATA"
Value of the 'screentip' Application Structure attribute for the associated APS.
region="CDATA"
Value of the 'region' Application Structure attribute for the associated APS.
viewcontext="CDATA"
Value of the 'viewcontext' Application Structure attribute for the associated APS.
visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.
interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.
paraEXT=""
paraEXT is a mechanism for adding additional child content (i.e., metadata) on the 'para'.
paraAttEXT=""
paraAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the 'para'.

See also the 'para' functional description in Section 3.

4.6 The 'subpara' element

The 'subpara' element of an XML companion file represents a CGM Application Structure of type 'subpara'. The corresponding 'subpara' is identifiable given its assigned 'apsid' attribute value.

```
    <!ENTITY % subparaEXT "" >
    <!ENTITY % subparaAttEXT "" >
    <!ELEMENT subpara ( linkuri %subparaEXT; )* >
    <!ATTLIST subpara apsid         ID           #REQUIRED
                      screentip     CDATA        #IMPLIED
                      region        CDATA        #IMPLIED
                      viewcontext   CDATA        #IMPLIED
                      visibility    ( on | off ) #IMPLIED
                      interactivity ( on | off ) #IMPLIED
                      %subparaAttEXT;
    >
```

Attribute definitions:
apsid="xml:id"
The unique identifier of a the Application Structure for the given WebCGM file.

screentip="CDATA"
Value of the 'screentip' Application Structure attribute for the associated APS.
region="CDATA"
Value of the 'region' Application Structure attribute for the associated APS.
viewcontext="CDATA"
Value of the 'viewcontext' Application Structure attribute for the associated APS.
visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.
interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.
subparaEXT=""
subparaEXT is a mechanism for adding additional child content (i.e., metadata) on the 'subpara'.
subparaAttEXT=""
subparaAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the 'subpara'.

See also the 'subpara' functional description in Section 3.

4.7 The 'linkuri' element

A 'linkuri' element of an XML companion file represents a WebCGM 'linkuri' Application Structure attribute. Contrary to other attributes, the 'linkuri' attribute is expressed as an element in the XML companion file. The corresponding Application Structure of this 'linkuri' is its parent element.

```
    <!ELEMENT linkuri EMPTY >
    <!ATTLIST linkuri uri      CDATA #REQUIRED
                      behavior CDATA #IMPLIED
                      desc     CDATA #IMPLIED >
```

Attribute definitions:
uri="CDATA"
The href of this 'linkuri' attribute. See section *Basic Data Types* for more information.
behavior="CDATA"
The behavior of this 'linkuri' attribute. See section *Basic Data Types* for more information.
desc="CDATA"
The title or description of this 'linkuri' attribute. See section *Basic Data Types* for more information.

See also the 'linkuri' functional description in Section 3.

4.8 The 'bindByName' element

A 'bindByName' element of an XML companion file is intended to correspond to one or more Application Structure in a CGM file. The common link between those Application Structures is that their 'name' or 'layername' attribute value corresponds to 'apstargetname'. See section *Relationship with XML companion file* for more information on the rules of mapping 'bindByName' attributes to WebCGM Application Structures.

```
    <!ENTITY % bindByNameEXT "" >
    <!ENTITY % bindByNameAttEXT "" >
    <!ELEMENT bindByName ( linkuri %bindByNameEXT; )* >
    <!ATTLIST bindByName apstargetname CDATA        #REQUIRED
                         screentip     CDATA        #IMPLIED
                         region        CDATA        #IMPLIED
                         viewcontext   CDATA        #IMPLIED
                         layerdesc     CDATA        #IMPLIED
                         visibility    ( on | off ) #IMPLIED
                         interactivity ( on | off ) #IMPLIED
                         %bindByNameAttEXT;
    >
```

Attribute definitions:
apstargetname="CDATA"
Name used to identify the corresponding Application Structure(s) for a given WebCGM file.
screentip="CDATA"
Value of the 'screentip' Application Structure attribute for the associated APS.
region="CDATA"
Value of the 'region' Application Structure attribute for the associated APS.
viewcontext="CDATA"
Value of the 'viewcontext' Application Structure attribute for the associated APS.
layerdesc="CDATA"
Value of the 'layerdesc' Application Structure attribute for the associated APS.
visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.
interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.
bindByNameEXT=""
bindByNameEXT is a mechanism for adding additional child content (i.e., metadata) on the APS.
bindByNameAttEXT=""
bindByNameAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the APS.

4.9 The 'bindById' element

The 'bindById' element of an XML companion file represents a CGM Application Structure of unknown type (some possibilities are: layer, grobject, para, subpara). The corresponding object is identifiable given its assigned 'apsid' attribute value. See section *Relationship with XML companion file* for more information on the rules of mapping 'bindById' attributes to WebCGM Application Structures.

```
    <!ENTITY % bindByIdEXT "" >
    <!ENTITY % bindByIdAttEXT "" >
    <!ELEMENT bindById ( linkuri %bindByIdEXT; )* >
    <!ATTLIST bindById apsid         ID           #REQUIRED
                       screentip     CDATA        #IMPLIED
                       region        CDATA        #IMPLIED
                       viewcontext   CDATA        #IMPLIED
                       layerdesc     CDATA        #IMPLIED
                       visibility    ( on | off ) #IMPLIED
                       interactivity ( on | off ) #IMPLIED
                       %bindByIdAttEXT;
    >
```

Attribute definitions:

apsid="xml:id"
The unique identifier of the Application Structure for the given WebCGM file.

screentip="CDATA"
Value of the 'screentip' Application Structure attribute for the associated APS.

region="CDATA"
Value of the 'region' Application Structure attribute for the associated APS.

viewcontext="CDATA"
Value of the 'viewcontext' Application Structure attribute for the associated APS.

layerdesc="CDATA"
Value of the 'layerdesc' Application Structure attribute for the associated APS.

visibility="on|off"
Value of the 'visibility' Application Structure attribute for the associated APS.

interactivity="on|off"
Value of the 'interactivity' Application Structure attribute for the associated APS.

bindByIdEXT=""
bindByIdEXT is a mechanism for adding additional child content (i.e., metadata) on the APS.

bindByIdAttEXT=""
bindByIdAttEXT is a mechanism for adding additional attributes (i.e., metadata) on the APS.

The complete WebCGM Companion File DTD follows.

```
<?xml version="1.0" encoding="UTF-16" ?>
<!DOCTYPE webcgm [
<!-- ================================================================= -->
<!-- This is the WebCGM XML Companion File DTD for use with            -->
<!-- WebCGM 2.0                                                        -->
<!-- ================================================================= -->
<!-- Original issue: September 2004                                    -->
<!--                                                                   -->
<!-- Revision history:                                                 -->
<!-- December 2004 (r1)                                                -->
<!-- Added xml version statement                                       -->
<!-- Corrected spelling of interactivity                               -->
<!-- Changed visibility/interactivity parameters from                  -->
<!-- CDATA to "on" or "off"                                            -->
<!-- January 2005 (r2)                                                 -->
<!-- Corrected linkrui to linkuri and subpara extensible               -->
<!-- attributes value                                                  -->
<!-- Added region and viewcontext attributes to                        -->
<!-- grobject, para, subpara, bindByIdd, and bindByName                -->
<!-- Change webcgm element attributes version and filename             -->
<!-- to #IMPLIED                                                       -->
<!-- ================================================================= -->
<!-- ================================================================= -->
<!-- Application specific entities                                     -->
<!-- Application groups define application specific attributes here    -->
<!-- and define the stubs for application specific elements that       -->
<!-- will be defined later in the DTD                                  -->
<!ENTITY % webcgmEXT ""                                                >
<!ENTITY % webcgmAttEXT ""                                             >
<!ENTITY % layerEXT ""                                                 >
<!ENTITY % layerAttEXT ""                                              >
<!ENTITY % grobjectEXT ""                                              >
<!ENTITY % grobjectAttEXT ""                                           >
<!ENTITY % paraEXT ""                                                  >
```

```
<!ENTITY % paraAttEXT ""                                                >
<!ENTITY % subparaEXT ""                                                >
<!ENTITY % subparaAttEXT ""                                             >
<!ENTITY % bindByIdEXT ""                                               >
<!ENTITY % bindByIdAttEXT ""                                            >
<!ENTITY % bindByNameEXT ""                                             >
<!ENTITY % bindByNameAttEXT ""                                          >

<!ELEMENT webcgm ( (layer | grobject | para | subpara |
                    bindById | bindByName %webcgmEXT;)* )               >
<ATTLIST  webcgm id       ID    #IMPLIED
                 version  '2.0' #FIXED
                 filename CDATA #IMPLIED
                 %webcgmAttEXT;                                         >

<!ELEMENT layer (%layerEXT;)*                                           >
<!ATTLIST layer apsid         ID           #REQUIRED
                layerdesc     CDATA        #IMPLIED
                visibility    ( on | off ) #IMPLIED
                interactivity ( on | off ) #IMPLIED
                %layerAttEXT;                                           >

<!ELEMENT grobject ( linkuri %grobjectEXT; )*                           >
<!ATTLIST grobject apsid         ID           #REQUIRED
                   screentip     CDATA        #IMPLIED
                   region        CDATA        #IMPLIED
                   viewcontext   CDATA        #IMPLIED
                   visibility    ( on | off ) #IMPLIED
                   interactivity ( on | off ) #IMPLIED
                   %grobjectAttEXT;                                     >

<!ELEMENT linkuri EMPTY                                                 >
<!ATTLIST linkuri uri      CDATA #REQUIRED
                  behavior CDATA #IMPLIED
                  desc     CDATA #IMPLIED                               >

<!ELEMENT para ( linkuri %paraEXT; )*                                   >
<!ATTLIST para apsid         ID           #REQUIRED
               screentip     CDATA        #IMPLIED
               region        CDATA        #IMPLIED
               viewcontext   CDATA        #IMPLIED
               visibility    ( on | off ) #IMPLIED
               interactivity ( on | off ) #IMPLIED
               %paraAttEXT;                                             >

<!ELEMENT subpara ( linkuri %subparaEXT; )*                             >
<!ATTLIST subpara apsid         ID           #REQUIRED
                  screentip     CDATA        #IMPLIED
                  region        CDATA        #IMPLIED
                  viewcontext   CDATA        #IMPLIED
                  visibility    ( on | off ) #IMPLIED
                  interactivity ( on | off ) #IMPLIED
                  %subparaAttEXT;                                       >

<!ELEMENT bindById ( linkuri %bindByIdEXT; )*                           >
<!ATTLIST bindById apsid         ID           #REQUIRED
```

```
                    screentip     CDATA        #IMPLIED
                    layerdesc     CDATA        #IMPLIED
                    region        CDATA        #IMPLIED
                    viewcontext   CDATA        #IMPLIED
                    visibility    ( on | off ) #IMPLIED
                    interactivity ( on | off ) #IMPLIED
                    %bindByIdAttEXT;                                    >

<!ELEMENT bindByName ( linkuri %bindByNameEXT; )*                       >
<!ATTLIST bindByName apstargetname CDATA        #REQUIRED
                     screentip     CDATA        #IMPLIED
                     layerdesc     CDATA        #IMPLIED
                     region        CDATA        #IMPLIED
                     viewcontext   CDATA        #IMPLIED
                     visibility    ( on | off ) #IMPLIED
                     interactivity ( on | off ) #IMPLIED
                     %bindByNameAttEXT;                                  >
<!--                                                                   -->
<!-- Define content models for application specific elements           -->
<!--                                                                   -->
]>
```

Back to top of chapter

---

5. WebCGM Document Object Model (DOM)

Contents

5.1 Overview

This DOM chapter defines a set of objects and interfaces for accessing and manipulating WebCGM documents. The functionality specified in this section is to allow script writers to manipulate WebCGM documents and to access information found in standard WebCGM XML companion files. The WebCGM 2.0 DOM API focuses its methods on: tree traversal, style changes, and providing access to metadata.

5.2 Relationship with XML DOM

Although inspired by the XML DOM specifications, the WebCGM DOM remains oriented towards WebCGM specific functionality. Since WebCGM uses a tree structure to group graphical primitives, it was therefore appropriate, to use a set of interfaces similar to the XML DOM Node, Element and Document interfaces. However, since WebCGM is expressed in a non-XML syntax, several changes had to be made to commonly known interfaces and methods in order to improve the user experience of WebCGM script writers.

The WebCGM DOM could almost be perceived as a 'readonly' DOM. Some interface methods allow users to change the visual appearance of Application Structures, but unlike the XML DOM specification, it does not allow for removal or insertion of WebCGMNodes into the object model. This constitute a significant difference between the specifications.

While WebCGM 1.0 offers interactivity support via hyperlinking and highlighting; the WebCGM 2.0 DOM takes it to the next level. The WebCGM 2.0 DOM borrows from the DOM Events specification, it introduces the concept of EventListeners and mouse Events in order to meet the requirements of WebCGM users.

5.3 Relationship with XML companion file

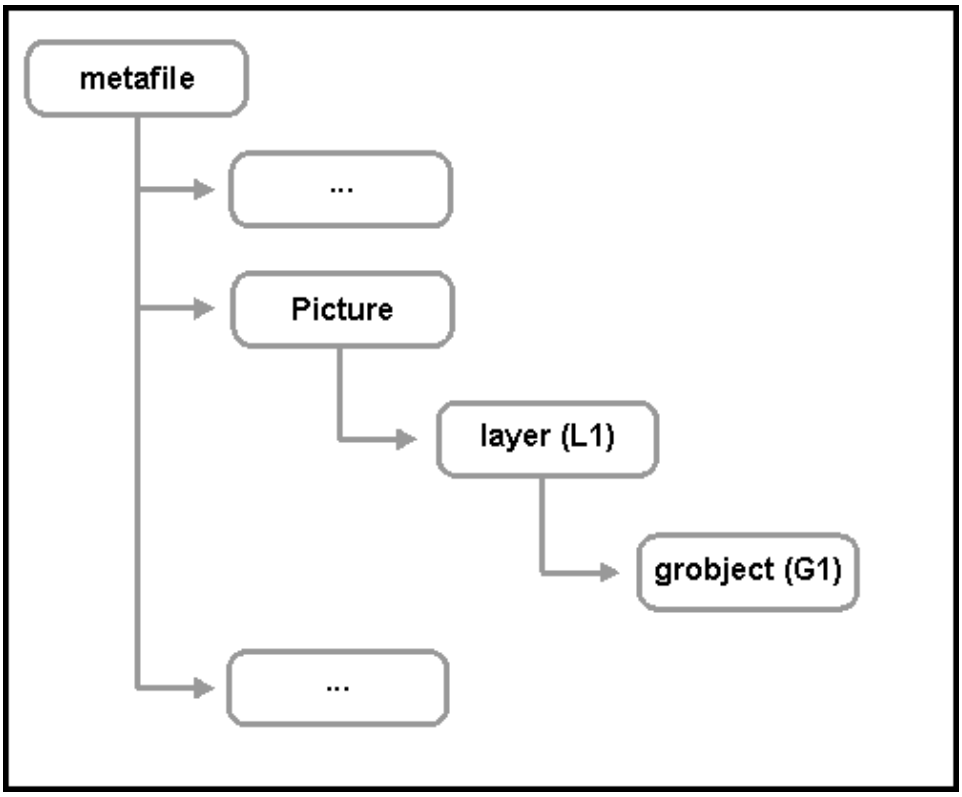
The WebCGM DOM is designed to provide access to XML metadata found in XML companion files. Practice has shown that some CGM illustrations are easier to maintain if some of the non graphical information remains outside the illustration. An example of such information could be; language sensitive tooltips. The WebCGM DOM can then be used to 'apply' the information from the XML companion file to the WebCGM document (see Example 5.3) . For more information on XML companion file syntax, please refer to ***Chapter 4, WebCGM XML Companion File***.

Another benefit of the XML companion file is to carry application specific data (or metadata) concerning a WebCGM illustration (see Example 4.2). This information is expressed using and elements in the XML companion file. The WebCGM DOM provides method for loading the XML metadata into the user agent's object model. Using the WebCGM DOM, a user can gain access to the metadata. Here is a detailed example to better illustrate the concept.

Example 5.1a: *This WebCGM document (expressed in clearText encoding) will be updated by an XML companion file.*

```
BEGMF 'example.cgm';
...
BEGPIC 'Picture 1';
...
BEGAPS 'L1' 'layer' STLIST;
  APSATTR 'layername' "14 1 'Standard layer'";
  BEGAPSBODY;
  BEGAPS 'G1' 'grobject' STLIST;
    BEGAPSBODY;
    LINE 210,265 210,200 300,200;
    LINE 300,200 300,265 210,265;
  ENDAPS;
ENDAPS;
...
ENDPIC;
...
ENDMF;
```

The in-memory tree representation of this illustration should be similar to the illustration found below. It is a simple tree structure with a root element WebCGMMetafile, one of the children of the root is a WebCGMPicture; the WebCGMPicture contains a Layer and the layer contains an Application Structure of type grobject.



Example 5.1b: *XML companion file to be 'applied' on example.cgm of example 5.1a.*

```
<webcgm id="example" xmlns:wiring="http://www.example.org">
  <grobject apsId="G1" screentip="A new screentip">
    <wiring:data color="rgb(255,0,0)"/>
  </grobject>
</webcgm>
```

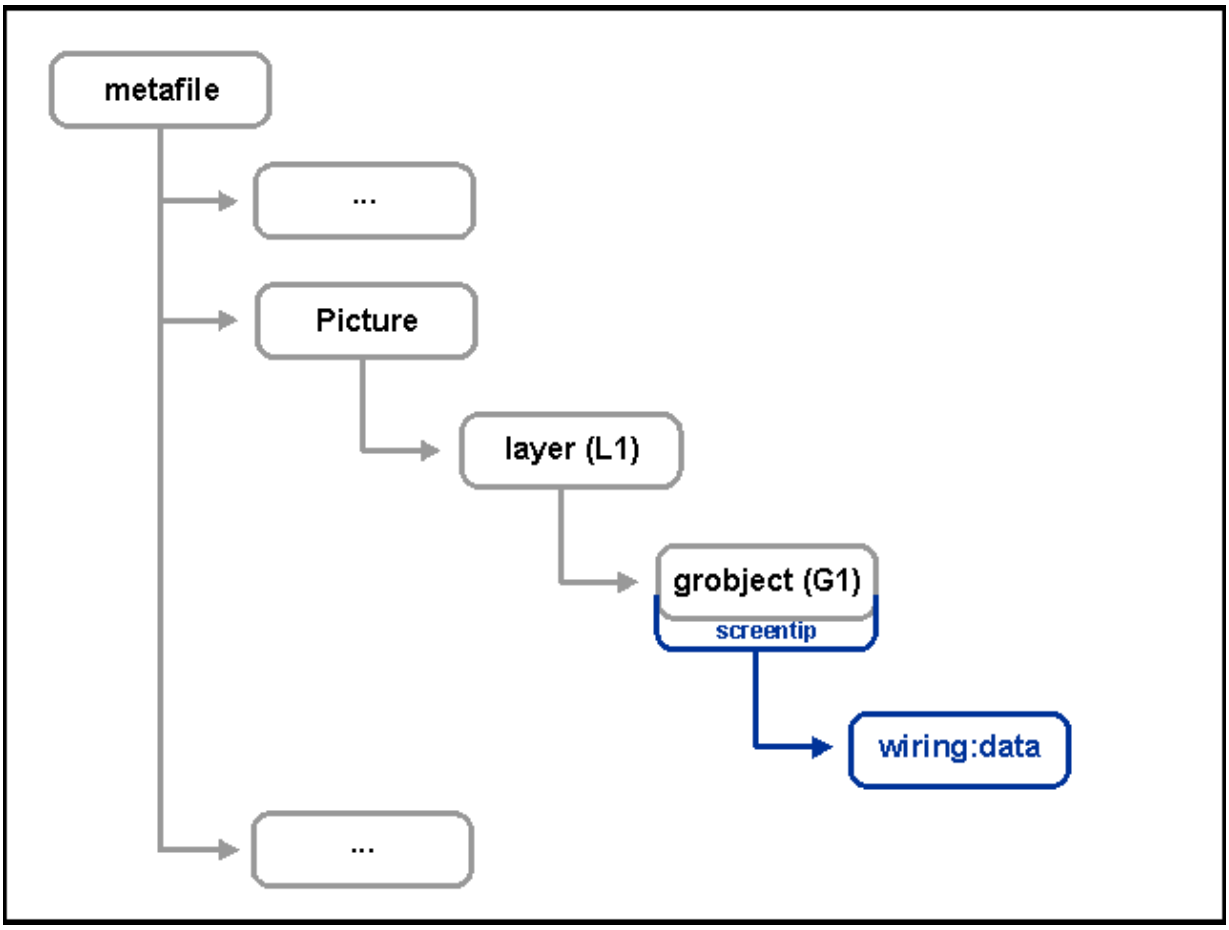
The WebCGM DOM provides methods for 'applying' an XML companion file, like the one shown in example 5.1, to a WebCGM document. A comforming user agent is expected to load and parse the XML companion file and possibly 'apply' updates from the XML companion file to the user agent's object model. A user may want to apply a companion file for the following reasons:

i) To replace standard Application Structure Attribute values present in the WebCGM instance with new values from the XML companion file.

ii) To supply standard Application Structure Attribute values to Application Structures which do not contain attribute values with values from the XML companion file.

iii) To add XML metadata to the user agent's object model to be retrieved at a later stage using WebCGM DOM APIs.

Once the user agent has loaded the XML companion file into its memory model, the tree should resemble this:



The overall set of rules that a user agent must follow when applying an XML Companion File on a WebCGM document is as follows:

1. Verify that root element is <webcgm>, else stop further processing and throw FILE_INVALID_ERR exception.
2. Process unknown attributes if any on root element, see below about **processing namespace attributes**.
3. Process all child elements using a depth-first algorithm.

More specific rules for **processing namespace attributes** are:

1. The target APS must be present in the CGM file, if that is not the case, all attributes of the current element are ignored.
2. If the attribute is not part of the base WebCGM DTD, it must be an extended namespace attribute, else the attribute is ignored.
3. An attribute that is already on the corresponding APS must be updated with the new attribute value.
4. In the case where an attribute with the same local name and namespace URI is already present on the APS, its prefix is changed to be the prefix part of the qualifiedName, and its value is changed to be the attribute value. If the attribute does not exist on the APS, the namespace attribute is appended onto the APS.

More specific rules for **processing child elements** are:

1. The target APS (the parent element) must be present in the CGM file, if that is not the case, all child elements of the current element are ignored. The target APS must not be of type 'grnode'. 'grnode' elements are not accessible via DOM calls.
2. If the element is not part of the base WebCGM DTD, it must be an extended namespace element. Namespace elements and their attributes are appended at the end of the target's list of child elements.
3. Elements that do belong in the WebCGM DTD are processed as follows:
    - namespace attributes are processed as specified above.
    - attributes relevant to this element, are updated on the APS.
    - other attributes are ignored.
4. If the element is a <linkuri>, the following rules apply:
    - If one or more 'linkuri' attribute(s) exists on the parent element, they are deleted. (i.e., it is not possible to add links to already existing links, the companion file has to contain all the links to take part of the 'linkuri' APS attribute).
    - The attributes of the <linkuri> element are combined to create a single WebCGM 'linkuri' APS attribute (as defined in WebCGM 1.0) on the parent element.
5. If the element is a <bindById>, only namespace attributes and attributes relevant to the target APS type are added to the target

(i.e., if the <bindById> has a 'screentip' attribute and the target APS is of type 'layer', the 'screentip' attribute will be ignored).

6. If the element is a <bindByName>, the user agent has to find all Application Structures that have a matching 'name' or layername' attribute. All found Application Structure are then subject to new attribute values (refer to the <bindById> description above).

5.4 Style attributes

5.4.1 Specified, computed, and actual values

WebCGM 2.0 user agents are required to support the inheritance model defined in this section. Once a user agent has loaded a document and constructed a document tree, it must assign, for every Application Structure in the tree, a value to every style attributes.

Very similar to the CSS model, the final value of a style attribute is the result of a four-step calculation: the value is determined through specification (the "specified value"), then resolved into a value that is used for inheritance (the "computed value"), then converted into an absolute value if necessary (the "used value"), and finally transformed according to the limitations of the local environment (the "actual value").

5.4.1.1 Specified values

User agents must first assign a specified value to each style attributes based on the following mechanisms (in order of precedence):

1. If the style attribute is assigned a value, use it.
2. Otherwise, if the style attribute is inherited and the Application Structure is not the root of the document tree, use the computed value of the parent Application Structure.
3. Otherwise use the style attributes's initial value. The initial value of each property is indicated in the style attribute's definition.

5.4.1.2 Computed values

Specified values are resolved to computed values after the document tree is create; for example relative units ('%') are computed to absolute values (i.e., RGB color or NVDC). Computing a value never requires the user agent to render the document.

When the specified value is not 'inherit', the computed value of a style attribute is determined as specified by the Computed Value line in the definition of the property. See the section on inheritance for the definition of computed values when the specified value is 'inherit'. The computed value exists even when the property doesn't apply, as defined by the 'Applies To' line. However, some style attributes may define the computed value of a style attribute for an Application Structure to depend on whether the style attributes applies to that Application Structure.

5.4.1.3 Used values

Computed values can be relative to each other; for example: To do. The used value is the result of taking the computed value and resolving these dependencies into a final absolute value used for the actual display.

5.4.1.4 Actual values

A used value is in principle the value used for rendering, but a user agent may not be able to make use of the value in a given environment. For example, a user agent may only be able to render borders with integer pixel widths and may therefore have to approximate the computed width, or the user agent may be forced to use only black and white shades instead of full colour. The actual value is the used value after any approximations have been applied.

5.4.2 Inheritance

Some values are inherited by the children of an element in the document tree, as described above. Each style attribute defines whether it is inherited or not. When inheritance occurs, elements inherit computed values. The computed value from the parent element becomes both the specified value and the computed value on the child.

5.4.2.1 The 'inherit' value

Each style attribute may also have a specified value of 'inherit',which means that, for a given element, the property takes the same

computed value as the style attribute for the Application Structure's parent. The 'inherit' value can be used to strengthen inherited values, and it can also be used on style attributes that are not normally inherited.Chapter on inheritance to be done.

5.5 Basic Data Types

To ensure interoperability, this specification specifies the following basic types used in various interfaces.

WebCGMString

A WebCGMString is a sequence of 16-bit units.

IDL Definition

```
    valuetype WebCGMString sequence<unsigned short>;
```

In WebCGM, like XML DOM Level 3, the UTF-16 encoding was chosen because of its widespread industry practice. For Java and ECMAScript, WebCGMString is bound to the String type because both languages also use UTF-16 as their encoding.The WebCGM DOM has many interfaces that imply string matching. For XML, string comparisons are case-sensitive and performed with a binary comparison of the 16-bit units of the WebCGMStrings.

Delimited WebCGMStrings

Some Application Structure attributes may have more than one specified value. Thus, some WebCGMStrings will in fact be delimited strings (ex: 'name' and 'linkuri' attributes). In the case of the 'linkuri' attribute, the value will always contain 3 * n strings (even if some components are optional), n representing the number of linkURI specified on the Application Structure. This restriction of 3 * n strings was put in place to ease the complexity of scripts aimed at manipulating delimited strings.

A delimited string conforms to the following notation:

```
DelimitedString ::= ListOfX | ListOfXX
ListX           ::= '"'Name'"' | '"'Name'"' Wsp ListX
ListXX          ::= "'"Name"'" | "'"Name"'" Wsp ListXX
Wsp             ::= (#x20 | #x9 | #xD | #xA)+
Name            ::= (ValidChar)* // Need a link for ValidChar.
```

Therefore, a multilink consisting of two links could be represented with the following delimited string: *'http://www.w3.org/' 'W3C' '_blank' 'http://www.cgmopen.org/' 'CGMOpen' '_self'.*

5.6 Coordinate values -- Normalized VDC (NVDC)

In a WebCGM instance, the representation of coordinates (VDC) is influenced by several CGM elements: VDC TYPE, VDC EXTENT, and SCALE MODE. WebCGM requires that SCALE MODE be 'metric', but places few other constraints. Therefore VDC (times some scale factor) are equivalent to millimeters, but otherwise the coordinate system could have a lot of variability: upper-left or lower-left origin, right-handed or left-handed, integer values or real values (floating or fixed), etc.

To simplify working with coordinates, the WebCGM DOM defines and uses a canonical, normalized coordinate system, Normalized VDC (NVDC).

NVDC units are millimeters, in a coordinate system whose origin corresponds to the lower left corner of the VDC extent, with the X axis pointing to the right, and the Y axis pointing up. The following examples illustrate the correspondence between NVDC and VDC values for several WebCGM instances.

Example 1: Simplest possible example, the VDC and the NVDC are identical

- VDC Type: Real
- VDC Extent (lower-left & upper-right corners): (0.,0.) (150.,100.)
- Scale Factor: 'metric', 1.0

The picture's VDC have lower-left origin, X increases to right, Y increases up, picture is 150 mm wide and 100 mm high. The NVDC are identical,: (0.,0.) for lower-left corner, (150.,100.) for upper-right corner. If `(x,y)` are VDC and `(x',y')` are NVDC, then:

- `x' = x`
- `y' = y`

Example 2: The VDC define an upper-left origin, and correspond to a U.S. paper size of 8.5x11.0 inches:

- VDC Type: Real
- VDC Extent (lower-left & upper-right corners): (0.,11.0) (8.5,0.)
- Scale Factor: 'metric', 24

In VDC space, the origin is the upper-left corner, X increases to right, Y increases down. In NVDC space, the lower-left corner coordinates (as always) are (0,0) and the upper right corner is (215.9,279.4). If `(x,y)` are VDC and `(x',y')` are NVDC, then:

- `x' = 25.4*x`
- `y' = 279.4 - 25.4*y`

Example 3: In the **general case**, if VDC Extent coordinates are `(xll, yll)`, `(xur, yur)`, and Scale Factor is 'metric', `s`, then `(x',y')` NVDC is derived from `(x,y)` VDC by:

- `x' = sign(xur-xll) * s * (x - xll)`
- `y' = sign(yur-yll) * s * (y - yll)`

5.7 Fundamental Interfaces

The interfaces within this section are considered fundamental, and must be fully implemented by all conforming implementations of the WebCGM DOM. The WebCGM DOM presents WebCGM documents as a hierarchy of **`WebCGMNode`** objects that also implement other, more specialized interfaces. Some of the node types may have child nodes of various types, and others are leaf nodes that cannot have anything below them in the WebCGM document structure. WebCGM has the following node types and children:

`WebCGMMetafile` -- contains a list of WebCGMPicture nodes.

`WebCGMPicture` -- may contain WebCGMAppStructures or XML metadata nodes.

`WebCGMAppStructure` -- may contain WebCGMAppStructures or XML metadata nodes.

`WebCGMAttr` -- no children.

The WebCGM DOM also specifies several other interfaces to facilitate access to WebCGM attributes. The **`GetWebCGMDocument`** interface is the medium between the host environment and the WebCGM functionality. The **`WebCGMNodeList`** interface allows to handle ordered lists of WebCGMNodes. The **`WebCGMEvent`** interface provides contextual information regarding mouse events. WebCGMNodeList objects in the DOM are live; that is, changes to the underlying document structure are reflected in all relevant NodeList objects. For example, if a DOM user gets a WebCGMNodeList object containing the children of an WebCGMAppStructure, then changes one of its children in the tree, all changes are reflected in the NodeList objects and in fact to all references to that Node in NodeList objects.

5.7.1 Exception WebCGMException

WebCGM operations only raise exceptions when an operation is impossible to perform.

IDL Definition

```
exception WebCGMException {
  unsigned short   code;
};

// ExceptionCode
const unsigned short      INDEX_SIZE_ERR                = 1;
const unsigned short      WEBCGMSTRING_SIZE_ERR         = 2;
const unsigned short      INVALID_CHARACTER_ERR         = 3;
const unsigned short      NO_DATA_ALLOWED_ERR           = 4;
const unsigned short      NO_MODIFICATION_ALLOWED_ERR   = 5;
const unsigned short      NOT_SUPPORTED_ERR             = 6;
const unsigned short      INVALID_ACCESS_ERR            = 7;
const unsigned short      FILE_NOT_FOUND_ERR            = 8;
const unsigned short      FILE_INCOMPATIBILITY_ERR      = 9;
```

Definition group ExceptionCode

An integer indicating the type of error generated

Defined Constants

INDEX_SIZE_ERR; if index or size is negative, or greater than the allowed value.

DOMSTRING_SIZE_ERR; if the specified range of text does not fit into a WebCGMString.

INVALID_CHARACTER_ERR; if an invalid or illegal character is specified, such as in an XML name.

NO_DATA_ALLOWED_ERR; if data is specified for a node which does not support data.

NO_MODIFICATION_ALLOWED_ERR; if an attempt is made to modify an object where modifications are not allowed.

NOT_SUPPORTED_ERR; if the implementation does not support the requested type of object or operation.

INVALID_ACCESS_ERR; if a parameter or an operation is not supported by the underlying object.

FILE_NOT_FOUND_ERR; if the reference document could not be accessed

FILE_INVALID_ERR; if the reference document was not well-formed or was in error.

5.7.2 Interface GetWebCGMDocument

Since WebCGM documents are often embedded within a host document such as XHTML, WebCGM user agents are required to implement the **GetWebCGMDocument** interface for the element which references the WebCGM document (e.g., the 'object' tag).

IDL Definition

```
interface GetWebCGMDocument {
  WebCGMMetafile             getWebCGMDocument ( ) raises ( WebCGMException );
};
```

Methods

getWebCGMDocument

Returns the WebCGMMetafile object for the referenced WebCGM document.

No parameters

Return value

WebCGMMetafile; The WebCGMMetafile object for the referenced WebCGM document.

Exceptions

WebCGMException; NOT_SUPPORTED_ERR: No WebCGMMetafile object is available.

Attributes

5.7.3 Interface WebCGMMetafile

The `WebCGMMetafile` interface is the entry point to the entire WebCGM document. The interface expose information regarding the metafile and provides access to the first WebCGMPicture of the WebCGM document.

IDL Definition

```
    interface WebCGMMetafile {
      readonly attribute WebCGMString  metafileDescription;
      readonly attribute WebCGMPicture firstPicture;
      readonly attribute WebCGMString  metafileID;
      readonly attribute usigned short metafileVersion;
               attribute WebCGMString  src;
    };
```

Attributes

metafileDescription of type WebCGMString, readonly

Returns the Metafile Descripton of the WebCGM document (e.g., "ProfileId:WebCGM,ProfileEd:1.0,Source:A software vendor,Date:20040602,ColourClass:monochrome"). The **metafileDescription** must contain the ProfileId: and the ProfileEd:, other information such as Source, ColourClass etc... is considered optional.

firstPicture of type WebCGMPicture, readonly

Returns the first WebCGMPicture element of the WebCGM document. Subsequent WebCGMPictures can be accessed using the WebCGMPicture interface. A WebCGM document contains at least one WebCGMPicture.

metafileID of type WebCGMString, readonly

Returns the Metafile Identifier (also known as the CGM ID).

metafileVersion of type unsigned short, readonly

Returns the Metafile Version of the WebCGM document.

src of type WebCGMString

The URI of the current document. On setting, the new document pointed to by the URI is loaded by the user agent. The user agent must fully parse the fragment identifier (if any) in the URI and execute the indicated behavior.

5.7.4 Interface WebCGMNode

The `WebCGMNode` interface is the base datatype of the WebCGM Document Object Model. The WebCGMNode object is the basis of several other interfaces; XMLElements and WebCGM specific elements (i.e., WebCGMAppStructure & WebCGMPicture). The

WebCGMNode interface specifies the attributes and methods to perform simple and generic tree traversal routines.

IDL Definition

```
interface WebCGMNode {
  const unsigned short PICTURE_NODE          = 1;
  const unsigned short APP_STRUCTURE_NODE    = 2;
  const unsigned short XML_METADATA_NODE     = 3;
  const unsigned short TEXT_NODE             = 4;
  const unsigned short ATTR_NODE             = 5;

  readonly attribute WebCGMString       nodeName;
  readonly attribute WebCGMString       nodeValue;
                                        // raises(WebCGMException) on retrieval

  readonly attribute unsigned short     nodeType;
  readonly attribute WebCGMNode         parentNode;
  readonly attribute WebCGMNodeList     childNodes;
  readonly attribute WebCGMNode         firstChild;
  readonly attribute WebCGMNode         lastChild;
  readonly attribute WebCGMNode         previousSibling;
  readonly attribute WebCGMNode         nextSibling;
  readonly attribute WebCGMPicture      ownerPicture;
  boolean                               hasChildNodes();
  boolean                               hasAttributes();
  boolean                               hasAttributeNS(in WebCGMString namespaceURI,
                                                       in WebCGMString localName);
  readonly attribute WebCGMNodeList     attributes;

  readonly attribute WebCGMString       namespaceURI;
  readonly attribute WebCGMString       prefix;

  readonly attribute WebCGMString       localName;

  WebCGMString                          getAttributeNS(in WebCGMString namespaceURI,
                                                       in WebCGMString localName);
  void                                  setAttributeNS(in WebCGMString namespaceURI,
                                                       in WebCGMString qualifiedName,
                                                       in WebCGMString value);
  WebCGMNodeList                        getElementsByTagNameNS(in WebCGMString
namespaceURI,
                                                               in WebCGMString
localName);
};
```

Definition group NodeType

An integer indicating which type of node this is.

Defined Constants:

PICTURE_NODE; the node is a WebCGMPicture.

APP_STRUCTURE_NODE; the node is a WebCGMAppStructure.

XML_METADATA_NODE; the node is XML companion information attached to a CGM element.

TEXT_NODE; the node contains character data.

ATTR_NODE; the node is a WebCGMAttr.

The values of nodeName and nodeValue vary according to the node type as follows:

Interface	nodeName	nodeValue
WebCGMAppStructure	WebCGMAppStructure type ("layer" | "grobject" | "para" | "subpara")	null
WebCGMAttr	WebCGMAttr.name	null
WebCGMPicture	"#picture"	null
Character Data	"#text"	content of the text node
XML Metadata	prefix + localName	null

Attributes

nodeName of type WebCGMString, readonly

The name of this node, depending on its type; see the table above.

nodeValue of type WebCGMString

The value of this node, depending on its type; see the table above.

Exceptions on setting

WebCGMException; NO_MODIFICATION_ALLOWED_ERR: Raised when the node is readonly and if it is not defined to be null.

Exceptions on retrieval

WebCGMException; DOMSTRING_SIZE_ERR: Raised when it would return more characters than fit in a WebCGMString variable on the implementation platform.

nodeType of type unsigned short, readonly

A code representing the type of the underlying object, see the table above.

parentNode of type WebCGMNode, readonly

The parent (immediate ancestor node of a node) of this node. All nodes, except WebCGMPicture may have a parent.

childNodes of type WebCGMNodeList, readonly

A WebCGMNodeList that contains all children of this node. If there are no children, this returns an empty WebCGMNodeList.

firstChild of type WebCGMNode, readonly

The first child of this node. If there is no such node, this returns null.

lastChild of type WebCGMNode, readonly

The last child of this node. If there is no such node, this returns null.

previousSibling of type WebCGMNode, readonly

The node immediately preceding this node. If there is no such node, this returns null.

nextSibling of type WebCGMNode, readonly

The node immediately following this node. If there is no such node, this returns null.

ownerPicture of type WebCGMPicture, readonly

The WebCGMPicture object associated with this node. When the node is a WebCGMPicture node, this returns null.

attributes of type WebCGMNodeList, readonly

A WebCGMNodeList containing the all attributes (WebCGM and namespaced) of this node or null if the WebCGMNode doesn't have any attributes.

namespaceURI of type WebCGMString, readonly

The namespace URI of this node (e.g., elementName xmlns="http://www.example.org/2004", returns "http://www.example.org/2004"). This is not a computed value that is the result of a namespace lookup based on an examination of the namespace declarations in scope.It is the namespace URI given at creation time. This returns null if the WebCGMNode is not of type XML_METADATA_NODE.

prefix of type WebCGMString, readonly

The namespace prefix of this node (e.g., foo:elementName, returns "foo"). This returns null if the WebCGMNode is not of type XML_METADATA_NODE or ATTR_NODE.

localName of type WebCGMString, readonly

Returns the local part of the qualified name of this node (e.g., foo:elementName, returns "elementName"). This returns null if the WebCGMNode is not of type XML_METADATA_NODE or ATTR_NODE.

Methods

hasChildNodes

Returns whether this node has any children.

No Parameters

Return Value

boolean; true if this node has any children, false otherwise.

No Exceptions

hasAttributes

Returns whether this node has any attributes.

No Parameters

Return Value

boolean; true if this node has any attributes, false otherwise.

No Exceptions

hasAttributeNS

Returns true when an attribute with a given local name and namespace URI is specified on this WebCGMNode, returns false otherwise.

No Parameters

Return Value

boolean; true if this node has any children, false otherwise.

No Exceptions

getAttributeNS

Returns the node attribute value by local name and namespace URI.

Parameters

namespaceURI of type WebCGMString

The namespace URI of the attribute to retrieve.

localName of type WebCGMString

The local name of the attribute to retrieve.

Return Value

WebCGMString; The WebCGMAttr value as a string, or the empty string if that attribute does not have a specified value.

No Exceptions

setAttributeNS

Adds a new attribute. If an attribute with that name is already present on the node, its value is changed to be that of the value parameter.

Parameters

namespaceURI of type WebCGMString

The namespace URI of the attribute to create or alter.

qualifiedName of type WebCGMString

The qualified name of the attribute to create or alter.

value of type WebCGMString

The value to set in string form.

No Return Value

Exception INVALID_CHARACTER_ERR: Raised if the specified qualified name contains an illegal character.

Exception NO_MODIFICATION_ALLOWED_ERR: Raised if this node is readonly.

getElementsByTagNameNS

Returns a WebCGMNodeList of all the descendant XML elements (companion information) with a given local name and namespace URI in the order in which they are encountered in a preorder traversal of the WebCGMNode tree.

Parameters

namespaceURI of type WebCGMString

The namespace URI of the XML elements to match on.

localName of type WebCGMString

The local name of the XML elements to match on.

Return Value

WebCGMNodeList; A list of matching XML element nodes.

No Exceptions

5.7.5 Interface WebCGMPicture

The `WebCGMPicture` interface allows for access to the application structures of the WebCGM document. It also specifies how to load and apply an XML companion file to a WebCGM document.

IDL Definition

```
interface WebCGMPicture : WebCGMNode {
  readonly attribute float           width;
  readonly attribute float           height;
  readonly attribute WebCGMString    pictid;

  void               addEventListener(in WebCGMString type,
                                      in WebCGMEventListener listener);
  void               removeEventListener(in WebCGMString type,
                                         in WebCGMEventListener listener));
  boolean            applyCcompanionFile(in WebCGMString fileURI);
  WebCGMNode         getAppStructureById(in WebCGMString apsId);
  WebCGMNodeList     getAppStructuresByName(in WebCGMString apsName);
  void               highlight(in WebCGMNodeList nodes,
                               in boolean state);
  void               setStyleAttr(in WebCGMString style,
                                  in WebCGMString value);
  void               clearStyleAttr(in WebCGMString style);
  void               reloadPicture();
};
```

Attributes

width of type float, readonly

Represents the WebCGMPicture width in millimeters. Please refer to Coordinate Values section for more information.

height of type float, readonly

Represents the WebCGMPicture height in millimeters. Please refer to Coordinate Values section for more information.

pictid of type WebCGMString, readonly

Represents the WebCGMPicture id, which is the id parameter in the BEGIN PICTURE element in the CGM document.

Methods

addEventListener

This method allows the registration of event listeners on the current WebCGMPicture node. If an WebCGMEventListener is added to the WebCGMPicture processing an event, it will not be triggered by the current actions. If multiple identical WebCGMEventListeners are registered on the same WebCGMPicture with the same parameters the duplicate instances are discarded. They do not cause the WebCGMEventListener to be called twice. Although all WebCGMEventListeners on the WebCGMPicture need to be triggered by any event which are guaranteed to be triggered by any event which is received by that WebCGMNode, no specification is made as to the order in which they will receive the event with regards to the other WebCGMEventListeners on the WebCGMNode.

Parameters

type of type WebCGMString

The event type for which the user is registering, (for example: "click", "mousemove").

listener of type WebCGMEventListener

The listener parameter takes an interface implemented by the user which contains the methods to be called when the event occurs.

No Return Value

No Exceptions

removeEventListener

This method allows the removal of event listeners on the current WebCGMPicture node. If an WebCGMEventListener is removed from the WebCGMPicture while it is processing an event, it will not be triggered by the current actions. WebCGMEventListeners can never be invoked after being removed. Calling removeEventListener with arguments which do not identify any currently registered WebCGMEventListener on the WebCGMPicture has no effect.

Parameters

type of type WebCGMString

Specifies the event type of the WebCGMEventListener being removed (for example: "click", "mousemove").

listener of type WebCGMEventListener

The WebCGMEventListener parameter indicates the WebCGMEventListener to be removed.

No Return Value

No Exceptions

applyCcompanionFile

The applyCompanionFile reads an XML companion file into the user agent's object model. If companion information is found in the companion file (in the form of namespace attributes and namespace children elements), the user agent will create new namespace application structures as children of existing WebCGM Application Structures within it's object model. This information will then be accessible using methods found on the WebCGMPicture, WebCGMAppStructure and WebCGMNode. Please refer to the interfaces. Please refer to the *Relationship with XML companion file* section for more detail.

Parameters

fileURI of type WebCGMString

The file name and location of the XML companion file to load and apply into the object model.

Return value

boolean; true if the implementation was able to load and parse the XML companion file into memory as requested; false otherwise.

Exceptions FILE_NOT_FOUND_ERR; if the reference document could not be accessed.

Exceptions FILE_INVALID_ERR; if the reference document was not well-formed or in error.

getAppStructureById

Returns the Application Structure whose ID is given by apsId. If no such Application Structure exists, returns null. Behavior is not defined if more than one element has this ID. Only WebCGM Application Structures may be retrieved using getAppStructureById, it does not retrieve arbitrary namespace elements (metadata elements).

Parameters

apsId of type WebCGMString

The unique id value for an Application Structure.

Return value

WebCGMNode; the matching Application Structure.

No Exceptions

getAppStructuresByName

Returns the list of Application Structures whose names are given by apsName in the order in which they are encountered in a preorder traversal of the WebCGMPicture tree. If no such Application Structures exists, returns null. Only WebCGM Application Structures may be retrieved using getAppStructuresByName.

Parameters

apsName of type WebCGMString

A non-unique name value for an Application Structure.

Return value

WebCGMNodeList; A WebCGMNodeList object containing all the matching Application Structure WebCGMNodes.

No Exceptions

highlight

Highlights a collection of Application Structures. WebCGM allows for highlighting of application structures using the URI fragment syntax. The exact method of highlighting is viewer dependent. The highlight method provides a way for WebCGM script writers to highlight application structures in the same way a URI fragment would. It also allows for highlighting of entire layers. Highlighting is not defined for WebCGMPicture nodes or XML Metadata nodes.

Parameters

nodes of type WebCGMNodeList

A WebCGMNodeList of APP_STRUCTURE_NODEs to highlight.

state of type boolean

A true value will highlight the nodes, whereas false will remove the highlight.

No Return value

No Exceptions

setStyleAttr

Set a style attribute at the picture level by name.

The following table describes in more detail each of the style attributes, their scopes and allowed values:

Style Attribute Name	WebCGMPicture level	APS level	Attribute value(s)	Example
background-color	yes	no	absolute RGB or relative intensity (0..100%)	"#000000" or "75%"
character-height	yes	yes	relative scale (> 0%)	"225%"
fill-color	yes	yes	absolute RGB or relative intensity (0..100%)	"#FF0000" or "75%"
intensity	yes	yes	intensity (0..100%)	"75%"
stroke-color	yes	yes	absolute RGB or relative intensity (0..100%)	"#FF0000" or "75%"
stroke-weight	yes	yes	relative scale (> 0%)	"225%"
text-color	yes	yes	absolute RGB or relative intensity (0..100%)	"#FF0000" or "75%"
text-font	yes	yes	WebCGMString	"Helvetica"
raster-intensity	yes	yes	relative intensity (0..100%)	"75%"
visibility	no	yes	WebCGMString	"visible" | "hidden"

Note: Descriptions of all style attributes have to be provided. stroke-color includes CGM attributes edge-color, line-color and marker-color. stroke-weight includes CGM attributes edge-weight and line-weight. RGB colors are expressed has hexadecimal values. Relative scale values are expressed as a non-negative number followed by a '%' unit, ex: "225%", the value can exceed 100%. Relative intensity values are expressed as a number followed by a '%' unit, ex: "75%", the value cannot exceed 100%.

absolute RGB colors are expressed using a hexadecimal representation for all three RGB channels, #RRGGBB. The following are two examples of colors expressed in hexadecimal representation: red is expressed as #FF0000 and cyan which uses both green and blue channels is expressed as #00FFFF. The shorthand hexadecimal notation is not supported in this specification.

background-color is the color rendering surface to which all elements cover, the entire default canvas will be white. A set value of #000000 will display a black background for all elements to render over.

character-height is the height of a character from the base line position together with the text-font white space between lines of text. The use of this style attribute value will increase or decrease the displayed character size by the value stated.

fill-color is the style attribute applied to a closed area inside the path of a shape. For example, if you draw a shape with a closed path, the fill color attribute will color the closed area.

intensity is a way to make the current color fade towards white. An intensity value of 0% will make the current application structure completely white while a value of 100% will keep the current color intact. The intensity equation is as follows:

```
normalizedNewRed = 1 - intensity * (1 - normalizedOldRed).

normalizedNewGreen = 1 - intensity * (1 - normalizedOldGreen).

normalizedNewBlue = 1 - intensity * (1 - normalizedOldBlue).
```

Here is an example of the computations when applying an intensity of 40% to the color orange #FFA500:

```
normalizedNewRed = 1 - 0.4 * (1 - 1) = 1.

normalizedNewGreen = 1 - 0.4 * (1 - 0.647) = 0.859.
```

```
normalizedNewBlue = 1 - 0.4 * (1 - 0) = 0.5.

The new color is %FFDB99.
```

Setting a relative intensity value is allowed on a number of style attributes, see table found above. The 'intensity' attribute however, represents a convinence attribute that controls the intensity value of the following four attributes: fill-color, stroke-color, text-color and raster-intensity.

stroke-color: In illustration, drawn lines are the strokes of a pen. The stroke-color is the applied color of the pen stroke for that line's presentation. This style attribute will apply an absolute or a relative intensity color change to the current color line value.

stroke-weight: The stroke-weight determines the thickness of the pen strokes for that line's presentation. The stroke-weight is centered on a drawn line of the path. This stroke-weight attribute will apply an absolute or a relative scale change to the current line value

text-color is the applied fill to text-font in the 'WebCGMString', the default fill text-color is black. This style attribute will apply an absolute or a relative intensity color change to the current color in the text string value.

text-font: TODO.

raster-intensity is the strength of which an image is visible and the strength of color or levels of grays that are displayed. The default relative intensity would be 100%.

Parameters

style of type WebCGMString

The name of the style attribute to modify.

value of type WebCGMString

The new value for the given style.

No Return value

No Exceptions

clearStyleAttr

Restores a style attribute to its original value (load time).

Parameters

style of type WebCGMString

The name of the style attribute. The special value "*" matches all style attributes.

No Return value

No Exceptions

reloadPicture

Notifies the user agent to immediately redraw the entire WebCGMPicture. The user agent will reload the WebCGMPicture while preserving the current user agent's zoom and pan level. The reloading of the WebCGMPicture also discards any existing companion information that may have been loaded into memory via the **applyCompanionFile** method.

No Parameters

No Return value

No Exceptions

5.7.6 Interface WebCGMAppStructure

The `WebCGMAppStructure` interface offers methods for setting and retrieving Application Structure attributes. The main methods for accessing Application Structure attributes are getAppStructureAttr and setAppStructureAttr, however; it is important to note that some attributes, like 'name' and 'linkuri', may have multiple values. In that case, a delimited string is returned.

The following table identifies which attribute values can be expressed as a delimited string. Each entry in the table points to the detailed description of the attribute, as it appears in WebCGM content.

APS Attribute Name	read/write	Delimited strings	Example
content	yes	no, single string	"car engine transmission"
interactivity	yes	no, single string	"on"
layerdesc	yes	no, single string	"This layer contains English instructions"
layername	readonly	no, single string	"English instructions"
linkuri	yes	yes, multiple values possible	"http://w3.org" "W3C" "_blank"
name	readonly	yes, multiple values possible	"firstName" "anotherName"
region	yes	yes, delimited	"1,0,0,100,100"
screentip	yes	no, single string	"This is a screentip"
viewcontext	yes	yes, delimited (two corner points)	"0,0,100,100"
visibility	yes	no, single string	"on"

The WebCGMAppStructure interface, like the WebCGMPicture provides methods for modifying style attribute interface, provides methods for modifying style attributes at the Application Structure level. For more information about available style attributes, refer to the Style Attributes Table.

IDL Definition

```
interface WebCGMAppStructure : WebCGMNode {
  readonly attribute WebCGMString  apsId;
  readonly attribute unsigned long nameCount;
  readonly attribute unsigned long linkuriCount;

  WebCGMString getAppStructureAttr(in WebCGMString name);
  void         setAppStructureAttr(in WebCGMString name,
                                   in WebCGMString value);
  void         removeAppStructureAttr(in WebCGMString name);
  void         setStyleAttr(in WebCGMString style,
                            in WebCGMString value);
  void         clearStyleAttr(in WebCGMString style);
};
```

Attributes

apsId of type WebCGMString, readonly

The unique identifier of the Application Structure.

nameCount of type unsigned long, readonly

Represents the number of 'name' attribute values present on this Application Structure.

linkuriCount of type unsigned long, readonly

Represents the number of 'linkuri' attribute values present on this Application Structure.

Methods

getAppStructureAttr

Retrieves an Application Structure attribute value by name. Please refer to the **Application Structure Attributes** table for more detailed information on retrievable and modifiable Application Structure attributes.

Parameters

name of type WebCGMString

The name of the Application Structure attribute to retrieve.

Return value

WebCGMString; the Application Structure attribute value as a string, or the empty string if that attribute does not have a specified value. The value may be a delimited string.

No Exceptions

setAppStructureAttr

Adds a new Application Structure attribute. If an attribute with that name is already present in the APS, its value is changed to be that of the value parameter. Please refer to the **Application Structure Attributes** table for more detailed information on retrievable and modifiable Application Structure attributes.

Parameters

name of type WebCGMString

The name of the Application Structure attribute to create or alter.

value of type WebCGMString

Value to set in string form. The value may be a delimited string.

No Return value

No Exceptions

removeAppStructureAttr

Removes an Application Structure attribute. Please refer to the **Application Structure Attributes** table for more detailed information on retrievable and modifiable Application Structure attributes.

Parameters

name of type WebCGMString

The name of the Application Structure attribute to remove.

No Return value

No Exceptions

setStyleAttr

Set a style attribute by name on the given Application Structure. Please refer to the **Style Attributes Table** for more detailed information on style attributes.

Parameters

style of type WebCGMString

The name of the style attribute to modify.

value of type WebCGMString

The new value for the given style.

No Return value

No Exceptions

clearStyleAttr

Restores a style attribute to its original value (load time).

Parameters

style of type WebCGMString

The name of the style attribute. The special value "*" matches all style attributes.

No Return value

No Exceptions

5.7.7 Interface WebCGMNodeList

The WebCGMNodeList interface provides the abstraction of an ordered collection of nodes. WebCGMNodeList objects in the WebCGM DOM are live. The index with the WebCGMNodeList starts at 0.

IDL Definition

```
    interface WebCGMNodeList {
      readonly   attribute unsigned long count;
      WebCGMNode           item(in unsigned long index);
      WebCGMNode           removeItem ( in unsigned long index )
                             raises( WebCGMException );
      WebCGMNode           appendItem ( in WebCGMString newItem )
                             raises( WebCGMException );
    };
```

Attributes

count of type unsigned long, readonly

The number of nodes in the list. The range of valid child node indices is 0 to count-1 inclusive.

Methods

items

Returns the index th item in the collection.

Parameters

index of type unsigned long

Index into the collection.

Return value

WebCGMNode; The node of the index th position in the WebCGMNodeList that is not a valid index., or null if that is not a valid index.

No Exceptions

removeItem

Removes an existing item from the list.

Parameters

index of type unsigned long

The index of the item which is to be removed. The first item is number 0.

Return value

WebCGMNode The removed item.

Exceptions

WebCGMException NO_MODIFICATION_ALLOWED_ERR: Raised when the list cannot be modified.

appendItem

Inserts a new item at the end of the list. If newItem is already in a list, it is removed from its previous list before it is inserted into this list.

Parameters

newItem of type WebCGMNode

The item which is to be inserted into the list. The first item is number 0.

Return value

WebCGMNode The inserted item.

Exceptions

WebCGMException NO_MODIFICATION_ALLOWED_ERR: Raised when the list cannot be modified.

5.7.8 Interface WebCGMAttr

The `WebCGMAttr` interface represents an attribute in a XML_METADATA_NODE, a PICTURE_NODE or a

APP_STRUCTURE_NODE.

IDL Definition

```
    interface WebCGMAttr: WebCGMNode {
      readonly attribute WebCGMString        name;
               attribute WebCGMString        value;
      readonly attribute WebCGMNode          ownerNode;
    };
```

Attributes

name of type WebCGMString, readonly

Returns the name of this attribute. If WebCGMNode.localName is different from null, this attribute is a qualified name.

value of type WebCGMString

On retrieval, the value of the attribute is returned as a string, see WebCGMNode.getAppStructureAttr(). On setting, is it equivalent to WebCGMNode.setAppStructureAttr().

Exceptions on setting

WebCGMException; NO_MODIFICATION_ALLOWED_ERR: Raised when the node is readonly and if it is not defined to be null.

ownerNode of type WebCGMNode, readonly

The Element node this attribute is attached to or null if this attribute is not in use.

5.7.9 Interface WebCGMEventListener

The WebCGMEventListener interface is the primary method for handling events. Users register their listener on the WebCGMPicture node with the addEventListener method.

IDL Definition

```
    interface WebCGMEventListener {
      void       handleEvent(in WebCGMEvent evt);
    };
```

Methods

handleEvent

This method is called whenever an event occurs of the type for which the WebCGMEventListener interface was registered.

Parameters

evt of type WebCGMEvent

The WebCGMEvent containing contextual information about the event.

No Return value

No Exceptions

5.7.10 Interface `WebCGMEvent`

The WebCGMEvent interface is used to provide contextual information about an event to the handler processing the event.

There exists three levels of interactivity in WebCGM:

- User-initiated actions such as a mouse click can be captured by the host environment and execute scripts.
- The user can initiate hyperlinks to Web pages or other WebCGM illustrations.
- User agent, users are able to zoom into and pan around WebCGM content.

This section also describes how a user agent processes the three different levels of interactivity.

When a mouse event occurs, the WebCGM user agent determines the target object of a mouse event. For the purposes of this discussion, "object" means Application Structure (APS). The target object is the topmost object whose relevant graphical content is under the mouse at the time of the event. An application structure of type 'grnode' or 'layer' cannot be a target of a mouse event. Instead, if the mouse pointer was over a 'grnode' when the event occurred; its closest ancestor object of type 'grobject', 'para' or 'subpara' will be designated as the target element. When an object is not displayed (i.e., 'visibility' attribute is set to off) or made non-interactive (i.e., 'interactivity' attribute is set to off), that object cannot be the target of mouse events.

The event is either initially dispatched to the target object, to the Picture, or not dispatched depending on the following:

- If there are no graphics objects whose relevant graphics content (see grobject, grnode, para or subpara for specifics) is under the mouse (i.e., there is no target element), the event is not dispatched.
- Otherwise, there is a target object. If there is an event handler at the Picture level with event capturing for the given event, then the event is dispatched to the Picture.
- Otherwise, if the target object has an appropriate event handler for the given event, the event is dispatched to the target object.
- Otherwise, the Picture is checked to see if it has an appropriate event handler. The event is dispatched to the Picture if one is found.
- Otherwise, the event is discarded.

The processing order for user interface events is as follows:

- Events handlers assigned to a WebCGMPicture get the event first via the potential event bubbling. If none of the activation event handlers take an explicit action (by invoking the preventDefault() WebCGM DOM method) to prevent further processing of the given event, then the event is passed on for:
- Cursor change, screentip and hyperlink processing. If a hyperlink is invoked in response to a user interface event, the hyperlink typically will disable further activation event processing (e.g., hyperlink to another Web page). If link processing does not disable further processing of the given event, then the event is passed on for:
- Document-wide event processing, such as user agent facilities to allow zooming and panning of a WebCGM document.

Since hyperlinks will in general change the context of a document it is more appropriate to allow explicit handlers to act on an event first and then process the hyperlink. The reverse order cannot guarantee that the script would get executed. Script writers should be made aware that this specification does not cover user agent event facilities such as zooming, panning or context menus. The mechanism to invoke such functionality will likely be different between vendors. Script writers are encouraged to become aware of those differences and thus, write highly interoperable WebCGM scripts.

IDL Definition

```
interface WebCGMEvent {
  readonly attribute WebCGMString     type;
  readonly attribute WebCGMNode       target;
  readonly attribute unsigned short   button;
  readonly attribute long             numPressed;
  readonly attribute float            clientX;
  readonly attribute float            clientY;
  readonly attribute boolean          ctrlKey;
  readonly attribute boolean          shiftKey;
  readonly attribute boolean          altKey;
  readonly attribute boolean          metaKey;

  void               preventDefault();
};
```

Attributes

type of type WebCGMString, readonly

The name of the event (case-insensitive). The name must be an XML name.

target of type WebCGMNode, readonly

Used to indicate the WebCGMNode (Application Structure) to which the event was originally dispatched.

button of type unsigned short, readonly

During mouse events caused by the depression or release of a mouse button, button is used to indicate which mouse button changed state. The values for button range from zero to indicate the left button of the mouse, one to indicate the middle button if present, and two to indicate the right button. For mice configured for left handed use in which the button actions are reversed the values are instead read from right to left.

numPressed of type long, readonly

Indicates the number of times a mouse button has been pressed and released over the same screen location during a user action. The attribute value is 1 when the user begins this action and increments by 1 for each full sequence of pressing and releasing. If the user moves the mouse between the mousedown and mouseup the value will be set to 0, indicating that no click is occurring.

clientX of type float, readonly

The horizontal coordinate at which the event occurred expressed in Normalized VDC.

clientY of type float, readonly

The vertical coordinate at which the event occurred expressed in Normalized VDC.

ctrlKey of type boolean, readonly

Used to indicate whether the 'ctrl' key was depressed during the firing of the event.

shiftKey of type boolean, readonly

Used to indicate whether the 'shift' key was depressed during the firing of the event.

altKey of type boolean, readonly

Used to indicate whether the 'alt' key was depressed during the firing of the event. On some platforms this key may map to an alternative key name.

metaKey of type boolean, readonly

Used to indicate whether the 'meta' key was depressed during the firing of the event. On some platforms this key may map to an alternative key name.

Methods

preventDefault

Calling preventDefault has the effect of cancelling the event. Any default action associated with the event will not occur.

No Parameters

No return value

WebCGM supports the following types of events:

click The click event occurs when the pointing device button is clicked. A click is defined as a mousedown and mouseup over the same screen location. The sequence of these events is: mousedown, mouseup, click. If multiple clicks occur at the same screen location, the sequence repeats with the detail attribute incrementing with each repetition. The Application Structure (if any) which was under the mouse pointer when clicked is populated in the WebCGMEvent.target property.

mousedown The mousedown event occurs when the pointing device button is pressed. The Application Structure (if any) which was under the mouse pointer when it was pressed down is populated in the WebCGMEvent.target property.

mouseup The mouseup event occurs when the pointing device button is released. The Application Structure (if any) which was under the mouse pointer when it was released populated in the WebCGMEvent.target property.

mouseover The mouseover event occurs when the pointing device is moved onto an Application Structure. The Application Structure that the mouse pointer moved over is populated in the WebCGMEvent.target property.

mouseout The mouseout event occurs when the pointing device is moved away from an Application Structure. The Application Structure that the mouse pointer moved away is populated in the WebCGMEvent.target property.

load The load event occurs when the WebCGM DOM implementation finishes loading all content within a WebCGM metafile.

unload The unload event occurs when the WebCGM DOM implementation removes a WebCGM metafile from a window or frame.

Back to top of chapter

---

6 WebCGM Profile

This section and its subsections are normative.

6.1 WebCGM Proforma

The following profile proforma (PPF) defines the WebCGM application profile for CGM files with a comparison to the ISO Model Profile as defined in ISO/IEC 8632:1999. The tables for the ISO Model Profile are duplicated here for reference and are intended to be accurate. In case of discrepancies, the Model Profile in ISO/IEC 8632:1999 shall take precedence. In the PPF, there are references such as 9.5.4.5, 5.5.11, and Annex I, etc. These are references to sections of the CGM:1999 text, from which this proforma is extracted. Other internal PPF references look like T16.13 and Attachment 26.4, which are references to table entries in the PPF itself.

6.2 Metafile Rules

Functionality	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.13.1	Same as Model Profile: **No**	
Encodings	Select 1 or more encodings: Binary **Yes**; Clear text **No**;	Select 1 or more encodings: Binary **Yes**; Clear text **Yes**;
T.13.2	Same as Model Profile: **Yes**	
Number of pictures	Number of pictures permitted in a metafile: minimum (>=0)? ***1.*** maximum (>=0 or no limit)? ***1.*** Other: None.	Number of pictures permitted in a metafile: minimum (>=0)? ***1.*** maximum (>=0 or no limit)? ***No limit.*** Other: ***None.***
T.13.3	Same as Model Profile: **Yes**	
Empty pictures		Are pictures allowed which have no graphical primitives? (yes/no) Yes. Other: ***None.***
T.13.4	Same as Model Profile: **Yes**	
Metafile size		Any restrictions on metafile size? ***None.*** Other: ***None.***

6.3 Multi-element Rules

Functionality	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.16.1	Same as Model Profile: **No**	
Colour References: 9.5.4.1	Select which rule applies to each metafile (choose 1): Either all colours or none shall be defined. **Yes**; All colours shall be defined. **No**; No colours shall be defined. **No**; Are colour indexes allowed to be redefined within a picture or metafile? (yes/no) ***Yes.*** Any restrictions on the number of distinct colours used within a picture or metafile? (Monochrome metafiles shall use at most two distinct colours.) ***None.*** Are conformance categories defined? (yes/no) ***Yes.*** If yes, specify. ***Monochrome and colour.*** Other: ***Grayscale is considered to be a special class of colour.***	Select which rule applies to each metafile (choose 1): Either all colours or none shall be defined. **Yes**; All colours shall be defined. **No**; No colours shall be defined. **No**; Are colour indexes allowed to be redefined within a picture or metafile? (yes/no) ***No.*** Any restrictions on the number of distinct colours used within a picture or metafile? (Monochrome metafiles shall use at most two distinct colours.) ***None.*** Are conformance categories defined? (yes/no) ***Yes.*** If yes, specify. ***3 categories: monochrome, grayscale, and colour***. Other: ***None.***
T.14.2	Same as Model Profile: **Yes**	
Line primitives - geometric degeneracies References: 9.5.4.3		Geometric degeneracies are: Permitted **Yes**; Prohibited **No**; If permitted, graphical meaning of the degeneracy: ***A line primitive element, whose entire locus is a single point, denotes a graphical dot which is a filled circle, with diameter equal to the current line width and colour equal to the current line colour.*** Other: ***None.***
T.14.3	Same as Model Profile: **Yes**	
Filled area primitives - geometric degeneracies References: 9.5.4.4		Geometric degeneracies are: Permitted **Yes**; Prohibited **No**; If permitted, graphical meaning of the degeneracy: ***A filled-area primitive element, whose entire locus is either a single point or a line has the following meaning:*** ***- If the locus of a filled-area primitive is a single point, then the meaning is a dot (which is a filled circle).*** ***- If the locus of a filled-area primitive is a non-***

		degenerate line segment, then the meaning is a line. ***The dot or line is displayed with the fill colour if EDGE VISIBILITY is 'off', unless INTERIOR STYLE is 'empty', in which case it is not rendered. If EDGE VISIBILITY is 'on', the interior treatment is the dot or line displayed in the fill colour, and then a dot or line superimposed with the current edge attributes.*** Other: ***None.***
T.14.4	Same as Model Profile: **No**	[redacted]
Graphical text strings References: 9.5.4.5	Minimum string length (bytes): ***0*** Maximum string length (bytes): ***254*** Any restrictions on the use of ISO/IEC 2022 switching controls? ***The C0 character NUL (code value) is permitted and has no effect. String parameters of graphical text shall contain no control character (7/8 bit codes: 1-31 and 128-159). ISO/IEC 2022 switching is not allowed in graphical text. A valid WebCGM metafile may use for graphical text only the character sets: the collection of four character sets which comprise ISO Latin1 and Symbol (see CHARACTER SET LIST); Unicode UTF-8; and UTF-16.*** Other: ***None.***	Minimum string length (bytes): 0. Maximum string length (bytes): ***254.*** Any restrictions on the use of ISO/IEC 2022 switching controls? ***Any character set used in the metafile which is accessed by ISO/IEC 2022 switching techniques shall be in the Character Set List (defined in this profile).*** Other: ***None.***
T.14.5	Same as Model Profile: **No**	[redacted]
Non-graphical text strings References: 9.5.4.6	Maximum string length (bytes): for type SF: ***254*** for type SF within type D: ***1024*** Format effectors and ESC: Permitted **Yes**; Prohibited **No**; Other C0 control codes (except NUL and ISO/IEC 2022 switching) are prohibited. Any limits on the set of acceptable character sets? ***The permitted character sets for non-graphical text are ISO Latin 1 (LHS & RHS), and UNICODE UTF-8, and Unicode UTF-16. Only one of these three shall be used throughout any particular WebCGM metafile instance. According to the CGM standard, the default SF character set, at the beginning of the 'metafile id' parameter of the BEGIN METAFILE element is ISO Latin 1. If the metafile is to use UTF-8 for SF parameters, then the following 4-octet ISO 2022 sequence shall occur as the first 4 octets of the 'metafile id' parameter:***	Maximum string length (bytes): for type SF: ***254.*** for type SF within type D: ***1024.*** Format effectors and ESC: Permitted **Yes**; Prohibited **No**; Other C0 control codes (except NUL and ISO/IEC 2022 switching) are prohibited. Any limits on the set of acceptable character sets? ***The permitted character sets are ISO 8859-1 LHS No.1 and ISO 8859-1 RHS No.1.*** Any restrictions on the use of ISO/IEC 2022 switching controls? ***Any character set used in the metafile which is accessed by ISO/IEC 2022 switching techniques shall be in the character set list (defined in this profile).***

	ESC 2/5 2/15 4/9 **If the metafile is to use UTF-16 for SF parameters, then the following 4-octet ISO 2022 sequence shall occur as the first 4 octets of the 'metafile id' parameter: ESC 2/5 2/15 4/12** **Otherwise, the use of ISO 2022 switching is prohibited in non-graphical text string.** ***NOTE: Section 6.3.4.5 of CGM:1999 allows the switching to UTF-8 (variable length multi-byte), and allows the use of 8-bit access to the ISO Latin 1 set.*** Any restrictions on the use of ISO/IEC 2022 switching controls? ***Any character set used in the metafile which is accessed by ISO/IEC 2022 switching techniques shall be in the character set list (defined in this profile).*** Other: ***See 3.1.1 for additional restrictions to the character repertoire for those WebCGM non-graphical strings which may be part of the WebCGM URI fragment.***	Other: ***None.***
T.14.6	Same as Model Profile: **Yes**	
Data record strings References: 9.5.4.7		Maximum string length (bytes) or state "no limit": ***32767.*** SDR-coding techniques must be used (see annex C.2.2). Other: ***None.***

6.4 Delimiter Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.15.1	Same as Model Profile: **Yes**	
BEGIN METAFILE END METAFILE [v1] References: 7.2.1 7.2.2		Element is: Required **Yes**; The *metafile identifier* shall follow the rules for non-graphical text, clause 9.5.4.6 and T.14.5. Other: ***None.***

9.5.4.6 T.14.5		
T.15.2	Same as Model Profile: **No**	
BEGIN PICTURE BEGIN PICTURE BODY END PICTURE [v1] References: 7.2.3 7.2.4 7.2.5 9.5.4.6 T.14.5	Element is: Required **Yes**; The *picture identifier* shall follow the rules for non-graphical text, clause 9.5.4.6 and T.14.5. Number of occurrences of these elements allowed in the metafile: ***1.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *picture identifier* shall follow the rules for non-graphical text, clause 9.5.4.6 and T.14.5. Number of occurrences of these elements allowed in the metafile: ***No limit.*** Other: ***None.***
T.15.3	Same as Model Profile: **No**	
BEGIN SEGMENT END SEGMENT [v2] References: 7.2.6 7.2.7	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneously defined segments (both global and local) at any point in the metafile: Any limits on the number of elements or restrictions on which elements compose a segment? Is there any meaning given to the *segment identifier* parameter? (yes/no) If yes, specify. (Meaning shall have no graphical effect.) Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneously defined segments (both global and local) at any point in the metafile: ***1024.*** Any limits on the number of elements or restrictions on which elements compose a segment? ***None.*** Is there any meaning given to the *segment identifier* parameter? (yes/no) ***No.*** If yes, specify. (Meaning shall have no graphical effect). Other: ***When global segments are specified in the Metafile Descriptor, all global segment definitions shall follow all other Metafile Descriptor elements. When segments are specified in the Picture Descriptor, all such segment definitions shall follow all other Picture Descriptor elements.***
T.15.4	Same as Model Profile: **No**	

BEGIN FIGURE END FIGURE [v2] References: 7.2.8 7.2.9	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the number of elements or restrictions on which elements comprise a figure definition: ***Maximum number of elements = 1024. No restrictions on which eligible elements may be included.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the number of elements or restrictions on which elements comprise a figure definition: ***Maximum number of elements = 128. No restrictions on which eligible elements may be included.*** Other: ***None.***
T.15.5	Same as Model Profile: **No**	
BEGIN PROTECTION REGION END PROTECTION REGION [v3] References: 7.2.10 7.2.11	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneously defined protection regions: ***1***. Maximum number of elements within each protection region: ***128***. Is there any meaning to the region index parameter other than as a unique identifier for each protection region? (yes/no) ***No***. If yes, specify. (Meaning shall have no graphical effect). Other: ***Region index is restricted to the value "1".***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneously defined protection regions: ***32.*** Maximum number of elements within each protection region: ***128.*** Is there any meaning to the region index parameter other than as a unique identifier for each protection region? (yes/no) No. If yes, specify. (Meaning shall have no graphical effect). Other: ***None.***
T.15.6	Same as Model Profile: **Yes**	
BEGIN COMPOUND LINE END COMPOUND LINE [v3] References: 7.2.12 7.2.13		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the number of elements and identity of elements comprising a path definition: ***Maximum number of elements is 128. No restrictions on which eligible elements may be included.*** Other: ***None.***
T.15.7	Same as Model Profile: **Yes**	

BEGIN COMPOUND TEXT PATH END COMPOUND TEXT PATH [v3] References: 7.2.14 7.2.15		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the number and identity of elements comprising a path definition: ***Maximum number of elements is 128. No restrictions on which eligible elements may be included.*** Other: ***None.***
T.15.8	Same as Model Profile: **No**	
BEGIN TILE ARRAY END TILE ARRAY [v3] References: 7.2.16 7.2.17	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of tiles in path direction: ***64*** Maximum number of tiles in line direction: ***64*** Maximum number of cells/tile in path direction: ***4096*** Maximum number of cells/tile in line direction: ***4096*** Limits on pel path: ***shall be 0.*** Limits on line progression: ***None.*** Limits on image offset: ***None.*** Other: ***Two types of raster images are allowed. A single (non-tiled) image has a maximum of 1,073,741,824 (1 giga, 32768**2) cells. A non-tiled image may exceed the 4096 cells/tile restriction specified for the tiled images. Tiled raster images are limited 64X64 tiles which are a maximum of 4096X4096 cells each. These tiled images are limited to a total of 1,073,741,824 (1 giga, 32768**2) cells (adjustment of maximum number of tiles and cells per tile are necessary to meet this requirement).***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of tiles in path direction: ***16.*** Maximum number of tiles in line direction: ***16.*** Maximum number of cells/tile in path direction: ***1024.*** Maximum number of cells/tile in line direction: ***1024.*** Limits on pel path: ***None.*** Limits on line progression: ***None.*** Limits on image offset: ***None.*** Other: ***None.***
T.15.9	Same as Model Profile: **No**	
BEGIN APPLICATION STRUCTURE BEGIN APPLICATION STRUCTURE BODY END APPLICATION	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the maximum number of defined structures within a picture:***None.*** Limits on the number and identity of elements comprising a structure: ***None.*** To the application structure identifier parameter, state the meaning:	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limits on the maximum number of defined structures within a picture: ***None.*** Limits on the number and identity of elements comprising a structure: ***None.*** To the application structure identifier parameter, state the meaning:

STRUCTURE [v4] References: 7.2.18 7.2.19 7.2.20	Assigned beyond being a unique identifier for the application structure: Is the inheritance flag parameter restricted? Yes/no: **Yes**. **The value of the inheritance flag is restricted to a value corresponding to "statelist"**. Other: **The value of the structure type parameter must be chosen from the list of valid structure types listed in the Section 3.2.1. Structures are sensitive to placement and are allowed according to the rules defined in Section 3.2.1, and placed in the metafile according to the content model defined in the XML DTD fragment in Section 3.3.**.	Assigned beyond being a unique identifier for the application structure: Is the inheritance flag parameter restricted? Yes/no: **No** Other: ***None.***

6.5 Metafile Descriptor Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.16.1	Same as Model Profile: **Yes**	[redacted]
METAFILE VERSION [v1] References: 7.3.1		Element is: Required **Yes**; Metafile versions permitted by this profile:***1, 2, 3, 4*** Other: ***None.***
T.16.2	Same as Model Profile: **No**	[redacted]
METAFILE DESCRIPTION [v1] References: 7.3.2 9.5.2.1 9.5.2.2 9.5.4.6 T.14.1 T.14.5	Element is: Required **Yes**; The *description* parameter shall follow the rules for non-graphical text, clause 9.5.4.6 and T.14.5. The substring within the SF parameter shall be of the form: "keyword:item", where the double quotes are part of the substring. Maximum number of occurrences of this element? ***1*** Profile identification (use keyword, "ProfileId:"): ***"ProfileId:WebCGM".*** Profile edition (use keyword, "ProfileEd:"): ***Refers to the approved version and revision of the specification that applies for this graphic. The Item associated with the keyword ProfileEd shall be n.m. For this WebCGM Edition: "ProfileEd:2.0".*** Additional information content: Metafile colour conformance class, source, and date items shall be encoded as substrings of the *description* parameter using the keywords: "ColourClass:", "Source:", and "Date:", respectively.	Element is: Required **Yes**; The *description* parameter shall follow the rules for non-graphical text, clause 9.5.4.6 and T.14.5. The substring within the SF parameter shall be of the form: "keyword:item", where the double quotes are part of the substring. Maximum number of occurrences of this element? ***Unlimited.*** Profile identification (use keyword, "ProfileId:"): ***"ProfileId:Model-Profile".*** Profile edition (use keyword, "ProfileEd:"): ***"ProfileEd:1".*** If the profile edition is not given, then the edition defaults to 1. Additional information content: Metafile colour conformance class, source, and date items shall be encoded as substrings of the *description* parameter using the keywords: "ColourClass:", "Source:", and "Date:", respectively.

	ColourClass: Required **Yes**; • Content: ***One of "ColourClass:monochrome" or "ColourClass:colour".*** Source? Required **No**; Permitted **Yes**; • Content: "***Source:supplier"*** Date? Required **No**; Permitted **Yes**; • Content: ***"Date:yyyymmdd"*** Other: ***Parameter strings are considered case insensitive.***	ColourClass: Required **Yes**; Permitted **No**; • Content: ***(One of: colour, grayscale, or monochrome.)*** Source? Required **Yes**; Permitted **No**; • Content: ***(Vendor, product, and version).*** Date? Required **Yes**; Permitted **No**; • Content shall be date of metafile generation. ***The form and content shall be in accordance with ISO 8601:1988.*** Other: ***None.***
T.16.3	Same as Model Profile: **No**	
VDC TYPE [v1] References: 7.3.3	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***None.*** Other: ***None.***	Element is: Required **Yes**; Permitted **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.16.4	Same as Model Profile: **Yes**	
INTEGER PRECISION [v1] References: 7.3.4		Element is: Required **No**; Permitted **Yes**; The parameter value of this element is encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.5	Same as Model Profile: **Yes**	
REAL PRECISION [v1] References: 7.3.5		Element is: Required **No**; Permitted **Yes**; The parameter value of this element is encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.6	Same as Model Profile: **Yes**	

INDEX PRECISION [v1] References: 7.3.6		Element is: Required **No**; Permitted **Yes**; The parameter value of this element is encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.7	Same as Model Profile: **Yes**	
COLOUR PRECISION [v1] References: 7.3.7		Element is: Required **No**; Permitted **Yes**; The parameter value of this element is encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.8	Same as Model Profile: **Yes**	
COLOUR INDEX PRECISION [v1] References: 7.3.8		Element is: Required **No**; Permitted **Yes**; The parameter value of this element is encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.9	Same as Model Profile: **No**	
MAXIMUM COLOUR INDEX [v1] References: 7.3.9	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is this element required to be a least upper bound? (yes/no) ***No.*** Any restrictions on the parameter values? • ***0-1 for monochrome metafiles.*** • ***0-255 for colour metafiles.*** Other: ***Grayscale is considered a special case of colour.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is this element required to be a least upper bound? (yes/no) ***No.*** Any restrictions on the parameter values? • ***0-1 for monochrome metafiles.*** • ***0-63 for grayscale metafiles.*** • ***0-255 for colour metafiles.*** Other: ***None.***
T.16.10	Same as Model Profile: **Yes**	

COLOUR VALUE EXTENT [v1] References: 7.3.10		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.16.11	Same as Model Profile: **Yes**	[redacted]
METAFILE ELEMENT LIST [v1] References: 7.3.11		Element is: Required **Yes**; Other: ***None.***
T.16.12	Same as Model Profile: **Yes**	[redacted]
METAFILE DEFAULTS REPLACEMENT [v1] References: 7.3.12		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is each occurrence of the MDR restricted to defining just one default? (yes/no) ***No.*** Additional restrictions may be specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.16.13	Same as Model Profile: **No**	[redacted]
FONT LIST [v1] References: 7.3.13 annex I	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element is required for all metafiles containing graphical text. Maximum number of fonts in the list: ***256*** All font indexes referenced in the metafile, including the default (nominally index 1) shall be defined in the FONT LIST element, with font name construction consistent with the rules of ISO/IEC 9541. List of ***recommended fonts:*** • ***Times-Roman*** • ***Times-Bold*** • ***Times-Italic*** • ***Times-BoldItalic*** • ***Helvetica*** • ***Helvetica-Bold*** • ***Helvetica-Oblique***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element is required for all metafiles containing graphical text. Maximum number of fonts in the list: ***64.*** All font indexes referenced in the metafile, including the default (nominally index 1) shall be defined in the FONT LIST element, with font name construction consistent with the rules of ISO/IEC 9541. List of permitted fonts: • ***Times-Roman*** • ***Times-Bold*** • ***Times-Italic*** • ***Times-BoldItalic*** • ***Helvetica*** • ***Helvetica-Bold*** • ***Helvetica-Oblique***

	• ***Helvetica-BoldOblique*** • ***Courier*** • ***Courier-Bold*** • ***Courier-Oblique*** • ***Courier-BoldOblique*** • ***Symbol*** NOTE - Theses font names are trademarked and some are proprietary and copyrighted. Times and Helvetica are registered trademarks of Allied Corporation, the owner of the copyright on the fonts of those names. Metric equivalents of the named fonts may be substituted by interpreters. Times is a serif font. Helvetica is a sans-serif font. Courier is a monospaced, serif font. The association of character code to glyph which shall be used for each of the fonts and the metrics of the named fonts are contained in clause I.2, annex I of CGM:1999. Other: ***The list of recommended fonts is intended to be a list of required minimum interpreter font capability and a recommended maximum font capability for generators. If other fonts are used, the FONT PROPERTIES and RESTRICTED TEXT elements are required. Font names are considered case insensitive.***	• ***Helvetica-BoldOblique*** • ***Courier*** • ***Courier-Bold*** • ***Courier-Oblique*** • ***Courier-BoldOblique*** • ***Symbol*** NOTE - Theses font names are trademarked and some are proprietary and copyrighted. Times and Helvetica are registered trademarks of Allied Corporation, the owner of the copyright on the fonts of those names. Metric equivalents of the named fonts may be substituted by interpreters. Times is a serif font. Helvetica is a sans-serif font. Courier is a monospaced, serif font. The association of character code to glyph which shall be used for each of the fonts and the metrics of the named fonts are contained in clause I.2, annex I. Other: ***None.***
T.16.14	Same as Model Profile: **No**	
CHARACTER SET LIST [v1] References: 7.3.14	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element is required for all metafiles containing graphical text. Maximum limit for the number of character sets in the character set list: ***6.*** Character sets shall be selected from the ISO Registry of Character Sets. This list may be extended by adding profile-defined character sets. List character sets: : ***"94-character G-set", 4/2 (ISO 8859-1 LH);*** ***"96-character G-set", 4/1 (ISO 8859-1 RH);*** ***"94-character G-set", 2/10 3/10 (Symbol LH);*** ***"94-character G-set", 2/6 3/10 (Symbol RH);***. ***"complete code", 4/9 (UTF-8)*** ***"complete code", 4/12 (UTF-16)*** If any of these character sets is of type "complete code", specify the content of the complete code and its associated sequence tail: ***Specified*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element is required for all metafiles containing graphical text. Maximum limit for the number of character sets in the character set list: ***4.*** Character sets shall be selected from the ISO Registry of Character Sets. This list may be extended by adding profile-defined character sets. List character sets: ***"94-character G-set", 4/2 (ISO 8859-1 LH);*** ***"96-character G-set", 4/1 (ISO 8859-1 RH);*** ***"94-character G-set", 2/10 3/10 (Symbol LH);*** ***"94-character G-set", 2/6 3/10 (Symbol RH).*** If any of these character sets is of type "complete code", specify the content of the complete code and its associated sequence tail: ***Not applicable.*** Other: ***None.***

T.16.15	Same as Model Profile: **No**	
CHARACTER CODING ANNOUNCER [v1] References: 7.3.15	Element is: Required **Yes**; Any restrictions on the parameter values? ***Value shall be 'basic 8-bit'.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***Values shall be 'basic 7-bit' and 'basic 8-bit'.*** Other: ***None.***
T.16.16	Same as Model Profile: **No**	
NAME PRECISION [v2] References: 7.3.16 Part 3, 8.3 Part 4, 7.2	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The parameter value of this element is coding dependent. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; The parameter value of this element is coding dependent. Other: ***None.***
T.16.17	Same as Model Profile: **Yes**	
MAXIMUM VDC EXTENT [v2] References: 7.3.17		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.16.18	Same as Model Profile: **No**	
SEGMENT PRIORITY EXTENT [v2] References: 7.3.18	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter values***.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.16.19	Same as Model Profile: **No**	

COLOUR MODEL [v3] References: 7.3.19	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of colour models? ***Shall be 1, 6, 7, or 8.*** Other: ***Values 6, 7, and 8 are the registered values for RGB-alpha, sRGB, and sRGB-alpha. sRBG and sRGB-alpha are deprecated and may be removed in a future version of WebCGM.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of colour models? ***None.*** Other: ***None.***
T.16.20	Same as Model Profile: **No**	
COLOUR CALIBRATION [v3] References: 7.3.20	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Calibration selection values permitted in accordance with the permitted model(s): If CYMK is permitted, minimum number of grid locations: Any restrictions on the number of colour lookup table entries, n? Any restrictions on the number of grid locations, m? If CYMK is permitted, algorithms for interpolation between grid locations? Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Calibration selection values permitted in accordance with the permitted model(s): ***Values 1..6, 9.*** If CYMK is permitted, minimum number of grid locations: ***3.*** Any restrictions on the number of colour lookup table entries, n? ***None.*** Any restrictions on the number of grid locations, m? ***None.*** If CYMK is permitted, algorithms for interpolation between grid locations? ***None.*** Other: ***None.***
T.16.21	Same as Model Profile: **No**	
FONT PROPERTIES [v3] References: 7.3.21	Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***The required parameters, when used, are INDEX, FONT FAMILY, POSTURE, WEIGHT, PROPORTIONATE WIDTH, DESIGN GROUP and STRUCTURE.*** Other: ***This element is required when a font is used that is not is the list of recommended fonts specified in the FONT LIST element. Parameter values of type SF are considered to be case insensitive***	Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***All defined index and enumerated values of all parameters shall be permitted.*** Other: ***None.***
T.16.22	Same as Model Profile: **No**	

GLYPH MAPPING [v3] References: 7.3.22	Required **No**; Permitted **No**; Prohibited **Yes**; Subset of AFII registered glyphs which may be referenced: Maximum number of glyphs which may be defined: Other: ***None.***	Required **No**; Permitted **Yes**; Prohibited **No**; Subset of AFII registered glyphs which may be referenced: ***None.*** Maximum number of glyphs which may be defined: ***8192.*** Other: ***None.***
T.16.23	Same as Model Profile: **Yes**	
SYMBOL LIBRARY LIST [v3] References: 7.3.23		Required **No**; Permitted **No**; Prohibited **Yes**; Libraries which may be accessed and their encoding rules: Maximum number of libraries which may be accessed: Other: ***NOTE - There are currently no registered symbol libraries.***
T.16.24	Same as Model Profile: **No**	
PICTURE DIRECTORY [v4] References: 7.3.24 9.5.4.6 T.14.5	Required **No**; Permitted **No**; Prohibited **Yes**; Follow the rules for non-graphical text strings for picture identifier parameter, clause 9.5.4.6 and T.14.5. Other: ***None.***	Required **No**; Permitted **Yes**; Prohibited **No**; Follow the rules for non-graphical text strings for picture identifier parameter, clause 9.5.4.6 and T.14.5. Other: ***None.***

6.6 Picture Descriptor Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.17.1	Same as Model Profile: **No**	
SCALING MODE [v1] References: 7.4.1	Element: Required **Yes**; Any restrictions on the parameter values? ***SCALING MODE shall be metric.*** Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***If SCALING MODE is metric then the 'metric scale factor' shall be positive.*** Other: ***None.***

T.17.2	Same as Model Profile: **Yes**	[redacted]
COLOUR SELECTION MODE [v1][v2] References: 7.4.2		Element: Required **No**; Permitted **Yes**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.17.3	Same as Model Profile: **Yes**	[redacted]
LINE WIDTH SPECIFICATION MODE [v1][v2] References: 7.4.3		Element: Required **No**; Permitted **Yes**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.17.4	Same as Model Profile: **Yes**	[redacted]
MARKER SIZE SPECIFICATION MODE [v1][v2] References: 7.4.4		Element: Required **No**; Permitted **Yes**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.17.5	Same as Model Profile: **Yes**	[redacted]
EDGE WIDTH SPECIFICATION MODE [v1][v2] References: 7.4.5		Element: Required **No**; Permitted **Yes**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.17.6	Same as Model Profile: **Yes**	[redacted]

VDC EXTENT [v1] References: 7.4.6		Element: Required **No**; Permitted **Yes**; Limits on the sense and orientation of the VDC space: ***None.*** Is zero-area VDC extent permitted? (yes/no) ***No.*** If yes, specify its meaning. Other: ***None.***
T.17.7	Same as Model Profile: **Yes**	
BACKGROUND COLOUR [v1] References: 7.4.7 9.5.4.1 T.14.1		Element: Required **No**; Permitted **Yes**; The *colour value* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Other: ***None.***
T.17.8	Same as Model Profile: **Yes**	
DEVICE VIEWPORT [v2] References: 7.4.8		Element: Required **No**; Permitted **No**; Prohibited **Yes**; Interaction of this element with environmental presentation directives: Meaning of this element if the specified value is inconsistent with the presentation device: Other: ***NOTE - This element is prohibited due to its device dependence.***
T.17.9	Same as Model Profile: **Yes**	
DEVICE VIEWPORT SPECIFICATION MODE [v2] References: 7.4.9		Element: Required **No**; Permitted **No** Prohibited **Yes**; Set of legal values: Other: ***NOTE - This element is prohibited due to its device dependence.***

T.17.10	Same as Model Profile: **Yes**	
DEVICE VIEWPORT MAPPING [v2] References: 7.4.10		Element: Required **No**; Permitted **No**; Prohibited **Yes**; Set of legal values: Other: ***NOTE - This element is prohibited due to its device dependence.***
T.17.11	Same as Model Profile: **No**	
LINE REP-RESENTATION [v2] References: 7.4.11 9.5.2.6 9.5.4.2 T.20.1	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneous bundle definitions: Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneous bundle definitions: ***20.*** Other: ***None.***
T.17.12	Same as Model Profile: **No**	
MARKER REP-RESENTATION [v2] References: 7.4.12 9.5.2.6 9.5.4.2 T.20.5	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneous bundle definitions: Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneous bundle definitions: ***20.*** Other: ***None.***
T.17.13	Same as Model Profile: **No**	

TEXT REP-RESENTATION [v2] References: 7.4.13 9.5.2.6 9.5.4.2 T.20.9	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneous bundle definitions: Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneous bundle definitions: ***20.*** Other: ***None.***
T.17.14	Same as Model Profile: **No**	
FILL REP-RESENTATION [v2] References: 7.4.14 9.5.2.6 9.5.4.2 T.20.21	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneous bundle definitions: Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneous bundle definitions: ***20.*** Other: ***None.***
T.17.15	Same as Model Profile: **No**	
EDGE REP-RESENTATION [v2] References: 7.4.15 9.5.2.6 9.5.4.2 T.20.26	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneous bundle definitions: Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneous bundle definitions: ***20.*** Other: ***None.***
T.17.16	Same as Model Profile: **Yes**	

INTERIOR STYLE SPECIFICATION MODE [v3] References: 7.4.16		Element: Required **No**; Permitted **Yes**; Any restriction on the parameter value? ***None.*** Other: ***None.***
T.17.17	Same as Model Profile: **Yes**	[redacted]
LINE AND EDGE TYPE DEFINITION [v3] References: 7.4.17		Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the number of definitions? ***Maximum of 32 line types shall be specified simultaneously.*** Any limits on the number of elements in a given definition? ***Number of values in the dash gap list shall not exceed 8.*** Any restrictions on the dash cycle repeat length? ***None.*** Any restrictions on complexity of definition to prevent degeneracies? ***None.*** Other: ***None.***
T.17.18	Same as Model Profile: **Yes**	[redacted]
HATCH STYLE DEFINITION [v3] References: 7.4.18		Element: Required **No**; Permitted **Yes**; Prohibited **No**; Limit on the number of hatch styles? ***Maximum of 32 hatch styles shall be specified simultaneously.*** Limit on the number of gaps in a given definition? ***Number of entries in the gap width list shall not exceed 8.*** Any limits on duty cycle length? ***None.*** Any restrictions on complexity of definition to prevent degeneracies? ***None.*** Any restrictions on the style indicator? ***None.*** Other: ***None.***
T.17.19	Same as Model Profile: **No**	[redacted]

GEOMETRIC PATTERN DEFINITION [v3] References: 7.4.19	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any limits on the number of geometric patterns defined? Any limits on the classes of primitives? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the number of geometric patterns defined? ***The maximum number of geometric patterns is 64.*** Any limits on the classes of primitives? ***None.*** Other: ***None.*** NOTE - The number of geometric patterns cannot exceed the number of segments.
T.17.20	Same as Model Profile: **No**	
APPLICATION STRUCTURE DIRECTORY [v4] References: 7.4.20 9.5.4.6 T.14.5	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Follow rules for non-graphical text strings *for application structure identifier* parameter, clause 9.5.4.6 and T.14.5. Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Follow rules for non-graphical text strings *for application structure identifier* parameter, clause 7.5.4.6 and T.14.5. Other: ***None.***

6.7 Control Elements

Element	**Specifications - WebCGM 2.0 Profile**	**Specifications - Model Profile**
T.18.1	Same as Model Profile: **Yes**	
VDC INTEGER PRECISION [v1] References: 7.5.1 Part 3, 8.5 Part 4, 7.4		Element is: Required **No**; Permitted **Yes**; The parameter values of this element are encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.18.2	Same as Model Profile: **Yes**	

VDC REAL PRECISION [v1] References: 7.5.2 Part 3, 8.5 Part 4, 7.4		Element is: Required **No**; Permitted **Yes**; The parameter values of this element are encoding dependent. Restrictions are specified in parts 3 and 4 of ISO/IEC 8632. Other: ***None.***
T.18.3	Same as Model Profile: **Yes**	
AUXILIARY COLOUR [v1] References: 7.5.3 9.5.4.1 T.14.1 D.4.4.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *auxiliary colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Other: ***None.***
T.18.4	Same as Model Profile: **Yes**	
TRANSPARENCY [v1] References: 7.5.4 9.5.7.9 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restriction on the parameter value? ***None.*** Other: ***None.***
T.18.5	Same as Model Profile: **Yes**	
CLIP RECTANGLE [v1] References: 7.5.5 D.4.4.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Meaning of boundary cases for: zero-area: Prohibited. area greater than VDC extent: Clipping shall be done to the intersection of CLIP RECTANGLE and VDC EXTENT.

		additional cases: None. Other: ***None.***
T.18.6	Same as Model Profile: **Yes**	[redacted]
CLIP INDICATOR [v1] References: 7.5.6		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.7	Same as Model Profile: **No**	[redacted]
LINE CLIPPING MODE [v2] References: 7.5.7 D.4.4.3	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter value? ***None.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.8	Same as Model Profile: **No**	[redacted]
MARKER CLIPPING MODE [v2] References: 7.5.8 D.4.4.3	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; P>Any restrictions on the parameter value? ***None.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.9	Same as Model Profile: **No**	[redacted]
EDGE CLIPPING MODE [v2] References: 7.5.9 D.4.4.3	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; P>Any restrictions on the parameter value? ***None.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.10	Same as Model Profile: **Yes**	[redacted]

NEW REGION [v2] References: 7.5.10		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element shall be permitted only if BEGIN FIGURE is permitted. Any restrictions on the number of occurrences? ***None.*** Other: ***None.***
T.18.11	Same as Model Profile: **No**	
SAVE PRIMITIVE CONTEXT [v2] References: 7.5.11	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Maximum number of simultaneously saved contexts: Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of simultaneously saved contexts: ***1024.*** Other: ***None.***
T.18.12	Same as Model Profile: **No**	
RESTORE PRIMITIVE CONTEXT [v2] References: 7.5.12	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; This element is permitted only if SAVE PRIMITIVE CONTEXT is permitted. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element is permitted only if SAVE PRIMITIVE CONTEXT is permitted. Other: ***None.***
T.18.13	Same as Model Profile: **No**	
PROTECTION REGION INDICATOR [v3] References: 7.5.13	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The values are restricted to: ***off, clip***. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element shall be permitted only if BEGIN PROTECTION REGION is permitted. Other: ***None.***
T.18.14	Same as Model Profile: **No**	

GENERALIZED TEXT PATH MODE [v3] References: 7.5.14	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***off, axis-tangential*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.15	Same as Model Profile: **Yes**	
MITRE LIMIT [v3] References: 7.5.15		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.18.16	Same as Model Profile: **Yes**	
TRANSPARENT CELL COLOUR [v3] References: 7.5.16 9.5.4.1 T14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *transparent cell colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter values? ***None.*** Other: ***None.***

6.8 Graphical Primitive Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.19.1	Same as Model Profile: **Yes**	
POLYLINE [v1] References: 7.6.1 T.14.2 D.2.21		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points or state "no limit": ***4096.*** Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***

T.19.2	Same as Model Profile: **Yes**	
DISJOINT POLYLINE [v1] References: 7.6.2 T.14.2 D.2.2.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points or state "no limit": ***4096.*** Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.3	Same as Model Profile: **Yes**	
POLYMARKER [v1] References: 7.6.3		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points or state "no limit": ***4096.*** Other: ***None.***
T.19.4	Same as Model Profile: **No**	
TEXT [v1] References: 7.6.4 9.5.4.5	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *string* parameter shall follow the rules for graphical text, clause 9.5.4.5. Is the 'not final' flag allowed: (yes/no) Other: ***Graphical text shall be represented by the Restricted Text element in this profile.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *string* parameter shall follow the rules for graphical text, clause 9.5.4.5. Is the 'not final' flag allowed: (yes/no) ***Yes.*** Other: ***None.***
T.19.5	Same as Model Profile: **Yes**	
RESTRICTED TEXT [v1] References: 7.6.5 9.5.4.5 T.25.4 D.4.5.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *string* parameter shall follow the rules for graphical text, clause 9.5.4.5. Is the 'not final' flag allowed: (yes/no) ***Yes.*** For[v1/2] metafiles, is the realization of RESTRICTED TEXT according to one of the standard or registered values for RESTRICTED TEXT TYPE? (yes/no)***Yes.*** If yes, specify. Boxed-cap, also see T.25.4 For [v3] metafiles, RESTRICTED TEXT TYPE shall be used if this element is used.

		Other: ***None.***
T.19.6	Same as Model Profile: **Yes**	[redacted]
APPEND TEXT [v1] References: 7.6.6. 9.5.4.5 D.4.5.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *string* parameter shall follow the rules for graphical text, clause 9.5.4.5. Other: ***None.***
T.19.7	Same as Model Profile: **Yes**	[redacted]
POLYGON [v1] References: 7.6.7 T.14.3 D.2.2.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points: *4096*. Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***
T.19.8	Same as Model Profile: **Yes**	[redacted]
POLYGON SET [v1] References: 7.6.8 T.14.3 D.2.2.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points: ***4096.*** Number of polygons in a set? ***No limit.*** Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***Each individual polygon within a set shall have at least 3 points.***
T.19.9	Same as Model Profile: **No**	[redacted]

CELL ARRAY [v1] References: 7.6.9 D.4.5.3	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limit for nx: ***32768*** Limit for ny: ***32768*** Limit for nx*ny:***1,073,741,824 ("1 giga", 32768**2).*** Are rotated and skewed cell arrays allowed? (yes/no) ***No.*** If yes, specify the graphical meaning. Other: ***Zero-area cell arrays are prohibited.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Limit for nx: ***2048.*** Limit for ny: ***2048.*** Limit for nx*ny: ***4194304.*** Are rotated and skewed cell arrays allowed? (yes/no) ***No.*** If yes, specify the graphical meaning. Other: ***Zero-area cell arrays are prohibited.***
T.19.10	Same as Model Profile: **Yes**	[redacted]
GENERALIZED DRAWING PRIMITIVE [v1] References: 7.6.10		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; List all the registered GDPs that are allowed: List all profile-defined GDPs that are allowed and attach complete description: Other:
T.19.11	Same as Model Profile: **Yes**	[redacted]
RECTANGLE [v1] References: 7.6.11 T.14.3 D.2.2.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: None.
T.19.12	Same as Model Profile: **Yes**	[redacted]
CIRCLE [v1] References: 7.6.12 T.14.3 D.2.2.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***

T.19.13	Same as Model Profile: **Yes**	[redacted]
CIRCULAR ARC 3 POINT [v1] References: 7.6.13 T.14.2 D.2.2.2 D.4.5.4		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.14	Same as Model Profile: **Yes**	[redacted]
CIRCULAR ARC 3 POINT CLOSE [v1] References: 7.6.14 T.14.3 D.2.2.2 D.4.5.5		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***
T.19.15	Same as Model Profile: **Yes**	[redacted]
CIRCULAR ARC CENTRE [v1] References: 7.6.15 T.14.2 D.2.2.2 D.4.5.6		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.16	Same as Model Profile: **Yes**	[redacted]

CIRCULAR ARC CENTRE CLOSE [v1] References: 7.6.16 T.14.3 D.2.2.2 D.4.5.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***
T.19.17	Same as Model Profile: **Yes**	[redacted]
ELLIPSE [v1] References: 7.6.17 T.14.3 D.2.2.2 D.4.5.9 D.4.5.10		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***
T.19.18	Same as Model Profile: **Yes**	[redacted]
ELLIPTICAL ARC [v1] References: 7.6.18 T.14.2 D.2.2.1 D.4.5.11		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.19	Same as Model Profile: **Yes**	[redacted]

ELLIPTICAL ARC CLOSE [v1] References: 7.6.19 T.14.3 D.2.2.2 D.4.5.12		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-area geometric degeneracies shall be as defined in T.14.3. Other: ***None.***
T.19.20	Same as Model Profile: **Yes**	
CIRCULAR ARC CENTRE REVERSED [v2] References: 7.6.20 T.14.2 D.2.2.1 D.4.5.8		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.21	Same as Model Profile: **Yes**	
CONNECTING EDGE [v2] References: 7.6.21 T.14.2 D.2.2.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; This element shall be permitted only if BEGIN/END FIGURE is permitted. Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.22	Same as Model Profile: **No**	

HYPERBOLIC ARC [v3] References: 7.6.22 T.14.2 D.2.2.1	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.23	Same as Model Profile: **No**	
PARABOLIC ARC [v3] References: 7.6.23 T.14.2 D.2.2.1	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.24	Same as Model Profile: **Yes**	
NON-UNIFORM B-SPLINE [v3] References: 7.6.24 T.14.2 D.2.2.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Set of spline orders: ***cubic spline.*** Maximum number of control points: ***4096.*** Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.25	Same as Model Profile: **Yes**	
NON-UNIFORM RATIONAL B-SPLINE [v3] References: 7.6.25		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Set of spline orders: ***cubic spline.*** Maximum number of control points: ***4096.*** Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***

T.14.2 D.2.2.1		
T.19.26	Same as Model Profile: **Yes**	
POLYBEZIER [v3] References: 7.6.26 T.14.2 D.2.2.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum number of points: ***4096.*** Any restrictions on the continuity indicator? ***None.*** Zero-length geometric degeneracies shall be as defined in T.14.2. Other: ***None.***
T.19.27	Same as Model Profile: **Yes**	
POLYSYMBOL [v3] References: 7.6.27 D.2.2.1		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Point list: Effect of a reference to a symbol index parameter which is not in the symbol library. Other: ***NOTE - This element is prohibited because SYMBOL LIBRARY LIST is prohibited.***
T.19.28	Same as Model Profile: **No**	
BITONAL TILE [v3] References: 7.6.28 D.2.2.1 D.4.5.13	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; List allowable compression types: ***0, 1, 2, 5, or 6.*** Requirements on row padding: ***None.*** Other: ***The values 0, 1 are deprecated and may be removed from a future version of WebCGM.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; List allowable compression types: ***Values 0..6.*** Requirements on row padding: ***None.*** Other: ***CCITT compression methods (T6 and T4) should be used with 1 bit cell colour precision and indexed colour.*** **Note - JPEG, LZW, and PNG have been registered in the ISO Registry of Graphical Items.**
T.19.29	Same as Model Profile: **No**	

TILE [v3] References: 7.6.29 D.2.2.1 D.4.5.13	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; List allowable compression types: ***2, 5, 6, 7, or 9*** Requirements on row padding: ***None.*** Other: ***The value 9 is the ISO registered value for compression method 0 of PNG. The values 0, 1, 2 are deprecated and may be removed from a future version of WebCGM.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; List allowable compression types: ***Values 0..6.*** Requirements on row padding? ***None.*** Other: ***CCITT compression methods (T6 and T4) should be used with 1 bit cell colour precision and indexed colour.*** **Note - JPEG, LZW, and PNG have been registered in the ISO Registry of Graphical Items.**

6.9 Attribute Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.20.1	Same as Model Profile: **No**	
LINE BUNDLE INDEX [v1] References: 7.7.1 9.5.4.2 D.4.6.1 T.17.11	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *line bundle index* parameter shall follow the rules for indexes, clause 7.5.4.2. For [v1] metafiles, allowable index values: For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *line bundle index* parameter shall follow the rules for indexes, clause 7.5.4.2. For [v1] metafiles, allowable index values: ***1..5.*** • **index 1 2 3 4 5** **line type 1 2 3 4 5** **line width 1.0 1.0 1.0 1.0 1.0** **line colour 1 1 1 1 1** For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***
T.20.2	Same as Model Profile: **No**	
LINE TYPE [v1] References: 7.7.2 9.4.17 D.4.6.2	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Select 1 or more of the following: • values 1..5: **Yes** • subset of registered values (attach list): 6..15: **Yes** • profile-defined values (attach complete description): **No** For [v3] metafiles, • negative values assigned by the LINE AND EDGE TYPE DEFINITION element. **Yes**;	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Select 1 or more of the following: • values 1..5; **Yes** • subset of registered values (attach list); **No** • profile-defined values (attach complete description); **No** For [v3] metafiles, • negative values assigned by the LINE AND EDGE TYPE DEFINITION element. **Yes**

	Other: ***Line types 6-15 are included in the Register of Graphical Objects. This register is available from the ISO SC24 Committee. See Section 4.18 about specific and generic line types.***	Other: ***None.***
T.20.3	Same as Model Profile: **Yes**	
LINE WIDTH [v1] References: 7.7.3 D.4.6.3		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is value zero allowed? (yes/no) ***Yes.*** If yes, specify its meaning.***Minimum available line width.*** Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.4	Same as Model Profile: **Yes**	
LINE COLOUR [v1] References: 7.7.4 9.5.4.1 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *line colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.5	Same as Model Profile: **No**	
MARKER BUNDLE INDEX [v1] References: 7.7.5 9.5.4.2 T.17.12 D.4.6.1	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *marker bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *marker bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: ***1..5.*** **index 1 2 3 4 5** **marker type 1 2 3 4 5** **marker width 1.0 1.0 1.0 1.0 1.0** **marker colour 1 1 1 1 1** For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***

T.20.6	Same as Model Profile: **Yes**	
MARKER TYPE [v1] References: 7.7.6 D.4.6.4		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Indicate one or more of the following restrictions: • values 1..5; **Yes** • subset of registered values (attach list); **No** • profile-defined values (attach complete description). **No** Other: ***None.***
T.20.7	Same as Model Profile: **Yes**	
MARKER SIZE [v1] References: 7.7.7 D.4.6.5		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is value zero allowed? (yes/no) ***Yes.*** If yes, specify its meaning. ***Minimum available size.*** Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.8	Same as Model Profile: **Yes**	
MARKER COLOUR [v1] References: 7.7.8 9.5.4.1 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *marker colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.9	Same as Model Profile: **No**	
TEXT BUNDLE INDEX [v1] References: 7.7.9 9.5.4.2 T.17.13	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *text bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. • For [v1] metafiles, allowable index values: For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *text bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: ***1..2.*** **index 1 2** **font index 1 1**

D.4.6.1		**text precision stroke stroke** **character expansion factor 1.0 0.7** **character spacing 0.0 0.0** **text colour 1 1** For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***
T.20.10	Same as Model Profile: **Yes**	
TEXT FONT INDEX [v1] References: 7.7.10 9.5.4.2 T.16.13		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Every referenced index shall refer to an entry in the FONT LIST (see T.16.13). Other: ***None.***
T.20.11	Same as Model Profile: **No**	
TEXT PRECISION [v1] References: 7.7.11	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***Value shall be 'stroke'.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.12	Same as Model Profile: **Yes**	
CHARACTER EXPANSION FACTOR [v1] References: 7.7.12 D.4.6.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is value zero allowed? (yes/no) ***No.*** If yes, state the meaning. Any restrictions on the parameter value? ***Values shall be restricted to the range 0.1..10.0*** Other: ***None.***
T.20.13	Same as Model Profile: **Yes**	

CHARACTER SPACING [v1] References: 7.7.13 D.4.6.8		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***Values shall be restricted to the range of -1.0..5.0.*** Other: ***None.***
T.20.14	Same as Model Profile: **Yes**	
TEXT COLOUR [v1] References: 7.7.14 9.5.4.1 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *text colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.15	Same as Model Profile: **Yes**	
CHARACTER HEIGHT [v1] References: 7.7.15 D.4.6.9		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is zero height allowed: (yes/no) ***Yes.*** If yes, state its meaning: Minimum available height. Any restrictions on the parameter? ***None.*** Other: ***None.***
T.20.16	Same as Model Profile: **Yes**	
CHARACTER ORIENTATION [v1] References: 7.7.16 D.4.6.10		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the following distortion aspects? rotation? **None.** skewing? **None.** mirroring? **None.** aspect ratio? **None.** Other: ***None.***

T.20.17	Same as Model Profile: **Yes**	[redacted]
TEXT PATH [v1] References: 7.7.17 D.4.6.11		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.18	Same as Model Profile: **Yes**	[redacted]
TEXT ALIGNMENT [v1] References: 7.7.18 D.4.6.12		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the horizontal and vertical alignment values? ***None.*** Any restrictions on the continuous horizontal and vertical alignment values? ***None.*** Other: ***None.***
T.20.19	Same as Model Profile: **No**	[redacted]
CHARACTER SET INDEX [v1] References: 7.7.19 9.5.4.2 T.16.14 T.16.22 D.4.6.13	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Every referenced index shall refer to an entry in the CHARACTER SET LIST. This includes implicit reference to the default index value. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Every referenced index shall refer to an entry in the CHARACTER SET LIST or GLYPH MAPPING. This includes implicit reference to the default index value. Other: ***None.***
T.20.20	Same as Model Profile: **No**	[redacted]

ALTERNATE CHARACTER SET INDEX [v1] References: 7.7.20 9.5.4.2 T.16.14 T.16.22 D.4.6.13	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Every referenced index shall refer to an entry in the CHARACTER SET LIST. This includes implicit reference to the default index value. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Every referenced index shall refer to an entry in the CHARACTER SET LIST or GLYPH MAPPING. This includes implicit reference to the default index value. Other: ***None.***
T.20.21	Same as Model Profile: **No**	
FILL BUNDLE INDEX [v1] References: 7.7.21 9.5.4.2 T.17.14 D.4.6.1	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *fill bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *fill bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: ***1..5.*** **index 1 2 3 4 5** **interior style hatch hatch hatch hatch hatch** **fill colour 1 1 1 1 1** **hatch index 1 2 3 4 5** **pattern index 1 1 1 1 1** For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***
T.20.22	Same as Model Profile: **Yes**	
INTERIOR STYLE [v1] References: 7.7.22 D.4.6.15		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; For 'hollow' interior style, line type and width of the bounding line: ***Solid line type and default line width.*** Any restrictions on the parameter value? ***None.*** Other: ***None.***

T.20.23	Same as Model Profile: **Yes**	
FILL COLOUR [v1] References: 7.7.23 9.5.4.1 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *fill colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.24	Same as Model Profile: **Yes**	
HATCH INDEX [v1] References: 7.4.18 7.7.24 6.7.4.3 D.4.6.16		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Select 1 or more of the following: • values 1..6: **Yes** • subset of registered values (attach list): **No** • profile-defined values (attach complete description): **No** For [v3] metafiles, • negative values assigned by the HATCH STYLE DEFINITION element. **Yes** Other: ***None.***
T.20.25	Same as Model Profile: **Yes**	
PATTERN INDEX [v1] References: 7.7.25 9.5.4.2		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *pattern index* parameter shall follow the rules for indexes, clause 9.5.4.2. Any restrictions on the parameter value? ***None***. Other: ***None***.
T.20.26	Same as Model Profile: **No**	

EDGE BUNDLE INDEX [v1] References: 7.7.26 9.5.4.2 T.17.15 D.4.6.1	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *edge bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *edge bundle index* parameter shall follow the rules for indexes, clause 9.5.4.2. For [v1] metafiles, allowable index values: ***1..5.*** • ***index 1 2 3 4 5*** **edge type 1 2 3 4 5** **edge width 1.0 1.0 1.0 1.0 1.0** **edge colour 1 1 1 1 1** For [v2/3] metafiles, any referenced bundle shall have an explicit representation definition. Other: ***None.***
T.20.27	Same as Model Profile: **Yes**	
EDGE TYPE [v1] References: 7.4.17 9.7.27 D.4.6.17		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Select 1 or more of the following: • values 1..5: **Yes** • subset of registered values (attach list): **No** • profile-defined values (attach complete description): **No** For [v3] metafiles, • negative values assigned by the LINE AND EDGE TYPE DEFINITION element. **Yes** Other: ***None.***
T.20.28	Same as Model Profile: **Yes**	
EDGE WIDTH [v1] References: 7.7.28 D.4.6.18		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Is value zero allowed? (yes/no) ***Yes.*** If yes, specify its meaning. Minimum available edge width. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.29	Same as Model Profile: **Yes**	

EDGE COLOUR [v1] References: 7.7.29 9.5.4.1 T.14.1		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; The *edge colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.30	Same as Model Profile: **Yes**	
EDGE VISIBILITY [v1] References: 7.7.30		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.31	Same as Model Profile: **Yes**	
FILL REFERENCE POINT [v1] References: 7.7.31		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? None. Other: None.
T.20.32	Same as Model Profile: **Yes**	
PATTERN TABLE [v1] References: 7.7.32		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Maximum size for nx: ***32.*** Allowable values for nx: ***8, 16, or 32.*** Maximum size for ny: ***32.*** Allowable values for ny: ***8, 16, or 32.*** Any restrictions on the number of pattern definitions? ***64.*** Any restrictions on allowable combinations of nx and ny? ***None.*** Any restrictions on the number of colours? ***None.*** Other: ***None.***

T.20.33	Same as Model Profile: **Yes**	
PATTERN SIZE [v1] References: 7.7.33 D.4.6.19		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Must pattern vectors be parallel to coordinate axes? (yes/no) ***Yes.*** If no, state the meaning of skewed or non-aligned patterns. Other:
T.20.34	Same as Model Profile: **No**	
COLOUR TABLE [v1] References: 7.7.34 9.5.4.1 T.14.1	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the length of colour list? ***Monochrome: 2, Colour: 256.*** Any restrictions on the index values? ***Index values shall not exceed the maximum colour index.*** Other: ***Grayscale metafiles are considered special cases of colour metafiles.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the length of colour list? ***Monochrome: 2, Grayscale: 64, Colour: 256.*** Any restrictions on the index values? ***Index values shall not exceed the maximum colour index.*** Other: ***None.***
T.20.35	Same as Model Profile: **No**	
ASPECT SOURCE FLAGS [v1] References: 7.7.35 D.4.6.20	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Are all ASF values to be the same: for the metafile? (yes/no) within each class (line, marker, text, fill, edge) of primitive? (yes/no) Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Are all ASF values to be the same: for the metafile? (yes/no) ***No.*** within each class (line, marker, text, fill, edge) of primitive? (yes/no) ***Yes.*** Other: ***None.***
T.20.36	Same as Model Profile: **No**	
PICK IDENTIFIER [v2] References: 7.7.36	Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter value? Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***

T.20.37	Same as Model Profile: **No**	
LINE CAP [v3] References: 7.7.37 9.5.7.5	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values for the line cap indicator? (choose 1 or both) **Yes**; values 1..4; **No**; subset of registered values (attach list). Any restrictions on the set of values for the dash cap indicator? (choose 1 or both) **Yes**; values 1..3; **No**; subset of registered values (attach list). Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values for the line cap indicator? (choose 1 or both) **Yes**; values 1..5; **No**; subset of registered values (attach list). Any restrictions on the set of values for the dash cap indicator? (choose 1 or both) **Yes**; values 1..3; **No**; subset of registered values (attach list). Other: ***None.***
T.20.38	Same as Model Profile: **Yes**	
LINE JOIN [v3] References: 7.7.38 9.5.7.5 T.26.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? (choose 1 or both) **Yes**; values 1..4; **No**; subset of registered values (attach list). Other: ***None.***
T.20.39	Same as Model Profile: **Yes**	
LINE TYPE CONTINUATION [v3] References: 7.7.39 9.5.7.5 T.26.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? ***1..4.*** Other: ***None.***
T.20.40	Same as Model Profile: **Yes**	

LINE TYPE INITIAL OFFSET [v3] References: 7.7.40		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.41	Same as Model Profile: **Yes**	
TEXT SCORE TYPE [v3] References: 7.7.41		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? (choose 1 or both) **Yes**; Values 1..4; **No**; Subset of registered values (attach list). Other: ***None.***
T.20.42	Same as Model Profile: **Yes**	
RESTRICTED TEXT TYPE [v3] References: 7.7.42 9.5.7.5 T.26.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? (choose 1 or both) **Yes**; Values 1..6; **No**; Subset of registered values (attach list). Algorithms for achieving restriction type? (attach) ***Not specified.*** Other: ***None.***
T.20.43	Same as Model Profile: **Yes**	
INTERPOLATED INTERIOR [v3] References: 7.7.43		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the number of stages? ***Maximum number of stages is 8.*** Any restrictions on the set of values? (choose 1 or both) **Yes**; Values 1..3; **No**; Subset of registered values (attach list). Other: ***None.***

T.20.44	Same as Model Profile: **Yes**	[redacted]
EDGE CAP [v3] References: 7.7.44 9.5.7.5 T.26.7	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values for the edge cap indicator? (choose 1 or both) **Yes**; values 1..4; **No**; subset of registered values (attach list). Any restrictions on the set of values for the dash cap indicator? (choose 1 or both) **Yes**; values 1..3; **No**; subset of registered values (attach list). Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values for the edge cap indicator? (choose 1 or both) **Yes**; values 1..5; **No**; subset of registered values (attach list). Any restrictions on the set of values for the dash cap indicator? (choose 1 or both) **Yes**; values 1..3; **No**; subset of registered values (attach list). Other: ***None.***
T.20.45	Same as Model Profile: **Yes**	[redacted]
EDGE JOIN [v3] References: 7.7.45 9.5.7.5 T.26.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? (choose 1 or both) **Yes**; values 1..4; **No**; subset of registered values (attach list). Other: ***None.***
T.20.46	Same as Model Profile: **Yes**	[redacted]
EDGE TYPE CONTINUATION [v3] References: 7.7.46 9.5.7.5 T.26.7		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the set of values? ***1..4.*** Other: ***None.***
T.20.47	Same as Model Profile: **Yes**	[redacted]

EDGE TYPE INITIAL OFFSET [v3] References: 7.7.47		Element is: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter value? ***None.*** Other: ***None.***
T.20.48	Same as Model Profile: **Yes**	
SYMBOL LIBRARY INDEX [v3] References: 7.7.48 9.5.4.2 T.16.23		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Every referenced index shall refer to an entry in the SYMBOL LIBRARY LIST (see T.16.23). Other: ***This element is prohibited because SYMBOL LIBRARY LIST is prohibited.***
T.20.49	Same as Model Profile: **Yes**	
SYMBOL COLOUR [v3] References: 7.7.49 9.5.4.1 T.14.1 T.16.23 D.4.6.21		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; The *symbol colour specifier* parameter shall follow the rules for colour, clause 9.5.4.1 and T.14.1. Any restrictions on the parameter value? Other: ***This element is prohibited because SYMBOL LIBRARY LIST is prohibited.***
T.20.50	Same as Model Profile: **Yes**	
SYMBOL SIZE [v3] References: 7.7.50 T.16.23		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Is value zero is allowed: (yes/no) If yes, specify its meaning. Any restrictions on the parameter value? Other: ***This element is prohibited because SYMBOL LIBRARY LIST is prohibited.***

T.20.51	Same as Model Profile: **Yes**	
SYMBOL ORIENTATION [v3] References: 7.7.51 T.16.23 D.4.6		Element is: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on rotation? Any restrictions on skewing? Any restrictions on mirroring? Any restrictions on distortion of aspect ratio? Other: ***This element is prohibited because SYMBOL LIBRARY LIST is prohibited***.

6.10. Escape Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.21.1	Same as Model Profile: **No**;	
ESCAPE [v1] References: 7.8.1	Element: Required **No**; Permitted **Yes**; Prohibited **No**; List all registered ESCAPEs that are allowed: • ***ESCAPE 22, Transparent Cell Colour [v1/v2] metafiles only.*** • ***ESCAPE 45, Alpha Transparency: The SDR parameter is encoded as a real value between 0.0 and 1.0, inclusively, and applies to all subsequent graphical primitives.*** List all profile-defined ESCAPEs that are allowed and attach complete description: Other: ***All ESCAPE element parameters shall be encoded as SDRs*** NOTE: Only registered ESCAPEs and profile-defined ESCAPEs shall be allowed in profiles.	Element: Required **No**; Permitted **Yes**; Prohibited **No**; List all registered ESCAPEs that are allowed: ***ESCAPE 22, Transparent Cell Colour [v1/v2] metafiles only***. List all profile-defined ESCAPEs that are allowed and attach complete description: ***None.*** Other: ***None***

6.11 External Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.22.1	Same as Model Profile: **No**	

MESSAGE [v1] References: 7.9.1	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Values of the *action required flag* parameter: • 'action' Permitted **No**; Prohibited **No**; (if permitted, specify the messages and actions taken) • 'no action' Permitted **No**; Prohibited **No**; Any restrictions on the length of the message string, other than those for type SF parameter? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Values of the *action required flag* parameter: • 'action' Permitted **No**; Prohibited **Yes**; (if permitted, specify the messages and actions taken) • 'no action' Permitted **Yes**; Prohibited **No**; Any restrictions on the length of the message string, other than those for SF parameter? ***None.*** Other: ***None.***
T.22.2	Same as Model Profile: **No**	
APPLICATION DATA [v1] References: 7.9.2	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Attach a syntactic and semantic description of all application data elements associated with this profile. Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Attach a syntactic and semantic description of all application data elements associated with this profile. Other: ***None.***

6.12 Segment Elements

Element	**Specifications - WebCGM 2.0 Profile**	**Specifications - Model Profile**
T.23.1	Same as Model Profile: **No**	
COPY SEGMENT [v2] References: 7.10.1 D.4.9.2	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Every segment identifier shall refer to a defined segment. Any limits on the segment transformation application value? Any restrictions on the nature of the transformation (e.g., permitting only isotropic transformations)? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Every segment identifier shall refer to a defined segment. Any limits on the segment transformation application value? ***None.*** Any restrictions on the nature of the transformation (e.g., permitting only isotropic transformations)? ***Non-singular.*** Other: *None.*
T.23.2	Same as Model Profile: **No**	

INHERITANCE FILTER [v2] References: 7.10.2	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any limits on the filter selection list? Any limits on the selection setting? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the filter selection list? ***None.*** An limits on the selection setting? ***None.*** Other: ***None.***
T.23.3	Same as Model Profile: **No**	[redacted]
CLIP INHERITANCE [v2] References: 7.10.3 D.4.9.2	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any limits on the parameter? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any limits on the parameter? ***None.*** Other: ***None.***
T.23.4	Same as Model Profile: **No**	[redacted]
SEGMENT TRANS-FORMATION [v2] References: 7.10.4	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the nature of the transformation (e.g., permitting only isotropic transformations)? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the nature of the transformation (e.g., permitting only isotropic transformations)? ***Non-singular.*** Other: ***None.***
T.23.5	Same as Model Profile: **No**	[redacted]
SEGMENT HIGHLIGHTING [v2] References: 7.10.5	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter values? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.23.6	Same as Model Profile: **No**	[redacted]

SEGMENT DISPLAY PRIORITY [v2] References: 7.10.6	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter values? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***None.*** Other: ***None.***
T.23.7	Same as Model Profile: **No**	
SEGMENT PICK PRIORITY [v2] References: 7.10.7	Element: Required **No**; Permitted **No**; Prohibited **Yes**; Any restrictions on the parameter values? Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Any restrictions on the parameter values? ***None.*** Other: ***None.***

6.13 Application Structure Elements

Functionality	**Specifications - WebCGM 2.0 Profile**	**Specifications - Model Profile**
T.24.1	Same as Model Profile: **No**	
APPLICATION STRUCTURE ATTRIBUTE [v4] References: 6.9 6.13.5 7.9.2 7.1.1	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Define the set of structure elements for use within application structures, and attach complete syntactic and semantic description: ***The set of attributes allowed is listed in Section 3 and are defined according to application structure type.*** Other: ***None.***	Element: Required **No**; Permitted **Yes**; Prohibited **No**; Define the set of structure elements for use within application structures, and attach complete syntactic and semantic description: ***None.*** Other: ***None.***

6.14 Generator Implementation Requirements

Functionality	**Specifications - WebCGM 2.0 Profile**	**Specifications - Model Profile**
T.25.1	Same as Model Profile: **Yes**	

Colour requirements References: 9.5.4.1 9.5.6.2.2		Colour mapping is: Permitted **Yes**; Prohibited **No**; Reduction of the number of colours? ***Not specified.*** NOTE - If mapping of application colours to metafile colour specifications is required, it is recommended that colour distance in the mapping be computed by the Euclidean metric in CIEXYZ space. Definition of mapping algorithms, metrics, and colour space? ***No specific colour mapping techniques or selection of metafile colour sets are defined.*** For [v1/2] metafiles, implicit colour calibration specifications? ***No specifications are defined.*** Other: ***None.***
T.25.2	Same as Model Profile: **Yes**	
Geometric accuracy and latitude References: 9.5.6.2.1		Accuracy and latitude for mapping application graphics to CGM graphical primitive elements: Accuracy and latitude for mapping application graphics to CGM graphical primitive elements: ***Generators shall produce a metafile whose graphical primitive elements match the application graphical primitives accurately to within ±0.1% of relative position within the VDC Extent box or ±1/2 pixel of the intended size, whichever is greater. Generators shall produce geometric size aspects of the primitives (e.g., text size, line width, and edge width) to within 1% of the intended size or ±1/2 pixel of the intended size, whichever is greater.*** This requirement shall apply to all graphical primitive elements, unless superseded by specific element requirements in this clause.
T.25.3	Same as Model Profile: **Yes**	
Text accuracy and latitude References: 9.5.6.2.3		Is text accuracy and latitude addressed? (yes/no) ***Yes***. If yes, specify. ***Metafile text specifications shall match the text of the application picture to within ±1% of relative to the intended size or ±1/2 pixel of the intended size, whichever is greater, for the placement and overall extent of each text string.***
T.25.4	Same as Model Profile: **No**	

Font substitution References: 9.5.6.2.4 annex I	Font substitution is: Permitted **Yes**; Prohibited **No**; Similarity of font visual characteristics? ***Substituted fonts shall be metrically equivalent or be controlled by the RESTRICTED TEXT element.*** Font metrics? ***Specified in ISO/IEC 8632:1999 Annex I for the core 13 fonts.*** Individual glyph metrics? ***Specified in ISO/IEC 8632:1999 Annex I for the core 13 fonts.*** Other: ***None.***	Font substitution is: Permitted **Yes**; Prohibited **No**; Similarity of font visual characteristics? ***Substituted fonts shall have similar visual characteristics (e.g., posture, weight, proportionate width).*** Font metrics? ***Specified in annex I.*** Individual glyph metrics? ***Specified in annex I.*** Other: ***None.***
T.25.5	Same as Model Profile: **Yes**	[redacted]
Preservation of primitives References: 9.5.6.3		Is preservation of graphical primitive elements addressed? (yes/no) ***No.*** If yes, specify allowable substitutions.
T.25.6	Same as Model Profile: **No**	[redacted]
Semantic latitude References: 9.5.6.4	Drawing priority and mode: ***Priority shall correspond to the metafile order (i.e., primitives occurring later in the file shall overlay primitives occurring earliest in the file). Mode shall be "replacement" mode.*** Clipping: ***Clipping shall be to the intersection of the clip rectangle, the VDC EXTENT, the device viewport, and the device view surface limits.*** Edge centreing: ***Edges shall be centred on the ideal mathematically-defined edge of the area*** Meaning of predefined line types and edge types: ***See Section 6.18 about specific and generic line types.*** Meaning of predefined hatch styles: ***See Section 6.19 about specific and generic hatch styles.*** Other: ***None.***	Drawing priority and mode: ***Priority shall correspond to the metafile order (i.e., primitives occurring later in the file shall overlay primitives occurring earliest in the file). Mode shall be "replacement" mode.*** Clipping: ***Clipping shall be to the intersection of the clip rectangle, the VDC EXTENT, the device viewport, and the device view surface limits.*** Edge centreing: ***Edges shall be centred on the ideal mathematically-defined edge of the area.*** Meaning of predefined line types and edge types: ***The exact on-off definitions for the predefined line types and edge types are not specified.*** Meaning of predefined hatch styles:***The inter-line spacing is not specified. Use the latitudes of annex D.4.6.16 for the angular directions.*** Other: ***None.***
T.25.7	Same as Model Profile: **Yes**	[redacted]

Error processing References: 9.5.6.5		Is error processing addressed? (yes/no) ***No.*** If yes, specify the action taken. Classification of error severity? Requirements for error recovery? Requirements for error reporting? Additional areas? Other: ***None.***
T.25.8	Same as Model Profile: **Yes**	
Reporting References: 9.5.6.6		Is reporting required? (yes/no) ***No.*** If yes, specify the action taken. Method and format of the reporting? • Requirement to report substitution, error, fallback behavior, mappings, or other behaviors? Additional areas? Other: ***None.***
T.25.9	Same as Model Profile: **Yes**	
Degeneracies References: 9.5.6.7 9.5.4.4 D.2 D.4		Is the generation of degenerate primitives addressed? (yes/no) ***No. The generation of degenerate primitives is not restricted.*** If yes, attach specifications. Other: ***None.***

6.15 Interpreter Implementation Requirements

Functionality	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.26.1	Same as Model Profile: **Yes**	

Number of pictures References: 9.5.7.2 T.13.2		If 0 pictures are permitted (see T.13.2), describe the interpreter behavior: ***Prohibited by T.13.2.***
T.26.2	Same as Model Profile: **Yes**	
Empty pictures References: 9.5.7.3 T.13.3		If permitted (see T.13.3), interpreter behavior: ***The graphical effect shall be one picture in the background colour.***
T.26.3	Same as Model Profile: **Yes**	
Colour requirements References: 9.5.4.1 9.5.7.4.2 9.5.4.5		Interpreters shall be classified as either monochrome, grayscale, or colour interpreters (depending on the colour capability of the interpreter), and shall meet the criteria in attachment 25.4 Conversions between different colour models shall be according to the conversions in annex G. Mapping of metafile colour to device components? ***If mapping (to fewer colour, or grayscale, or monochrome) is required for RGB metafiles, the recommendations of annex D.3.2 shall be used.*** For [v1/2] metafiles, implicit colour calibration specifications? ***No specifications are defined.*** Other: ***None.***
T.26.4	Same as Model Profile: **Yes**	
Geometric accuracy and latitude References: 9.5.7.4.1		Accuracy and latitude for placement and realization of geometric aspects when geometric primitive elements are rendered. ***Interpreters shall render graphical primitive elements accurately to within ± 0.1% of relative position within the VDC Extent box or ±1/2 of the pixel resolution of the output device, whichever is greater. Interpreters shall render the geometric size aspect of primitives (e.g., text size, line width, and edge width) to within 1% of the intended size or ±1/2 pixel of resolution of the output device, whichever is greater.*** This requirement shall apply to all graphical primitive elements, unless superseded by specific element requirements in this clause.

T.26.5	Same as Model Profile: **Yes**	
Text rendering References: 9.5.7.4.3		Is text accuracy and latitude addressed? (yes/no) ***Yes.*** If yes, specify. Interpreter-rendered text shall match the text specification of the metafile to within 1% relative to the intended size or ±1/2 pixel of resolution of the output device, whichever is greater, for the placement and overall extent of each text string. Is precision of text rendering is addressed? (yes/no) ***Yes.*** If yes, specify interpreter action. Interpreters shall render text using 'stroke' precision, regardless of the actual value of the TEXT PRECISION of the metafile.
T.26.6	Same as Model Profile: **No**	
Font substitution References: T.16.13 9.5.7.4.4 annex I	Font substitution is: Permitted **Yes**; Prohibited **No**; If prohibited, use the font as specified in the FONT LIST. If permitted, include a reference set of font and glyph metrics which correspond to the canonical instances of the substitutable font. See the FONT LIST element and annex I CGM:1999. Are substitution methods, latitudes, and constraints addressed? (yes/no) **No** If yes, specify: Similarity of font visual characteristics? ***Substituted fonts shall be metrically equivalent or be controlled by the RESTRICTED TEXT element.*** Font metrics? Substituted fonts shall have similar metrics to the fonts specified in the metafile. Individual glyph metrics? Specified in ISO/IEC 8632:1999 Annex I for the core thirteen fonts. Additional areas? None. Other: ***None.***	Font substitution is: Permitted **Yes**; Prohibited **No**; If prohibited, use the font as specified in the FONT LIST. If permitted, include a reference set of font and glyph metrics which correspond to the canonical instances of the substitutable font. See the FONT LIST element and annex I. Are substitution methods, latitudes, and constraints addressed? (yes/no) ***Yes.*** If yes, specify: Similarity of font visual characteristics? ***Substituted fonts shall have similar visual characteristics to the fonts specified in the metafile*** Font metrics? Substituted fonts shall have similar metrics to the fonts specified in the metafile. Individual glyph metrics? As specified in annex I. Additional areas? None. Other: ***None.***
T.26.7	Same as Model Profile: **No**	

Semantic latitude References: 9.5.7.5 T.20.37 T.20.38 T.20.39 T.20.42 T.20.44 T.20.45 T.20.46	Drawing priority and mode: ***Priority shall correspond to the metafile order (i.e., primitives occurring later in the file shall overlay primitives occurring earliest in the file. Mode shall be "replacement" mode.)*** View surface clearing at picture start: ***Surface will be cleared upon the occurrence of BEGIN PICTURE BODY, except as specified elsewhere in this profile.*** Clipping: ***When CLIP INDICATOR is 'off', clipping shall be to the intersection of the device viewport and the device view surface limits. When CLIP INDICATOR is 'on', clipping shall be to the intersection of the clip rectangle, the VDC EXTENT, the device viewport, and the device view surface limits*** Edge centreing: ***Edges shall be centred on the ideal mathematically-defined edge of the area.*** Meaning of predefined line types and edge types: ***See Section 6.18 about specific and generic line types.*** Meaning of predefined hatch styles: ***See Section 6.19 about specific and generic hatch styles.*** In the absence of a LINE/MARKER/TEXT/EDGE CLIPPING MODE element, the interpreter treatment of LINE/MARKER/TEXT/EDGE CLIPPING MODE shall be: In the style of one specific parameter value, from the set of standardized values. **YES.** Specify which one: **SHAPE** In the style of any of the specific parameter values, from the set of standardized values. **NO** For [v1/v2] metafiles, text restriction method for RESTRICTED TEXT elements, chosen from the set of standard and registered styles of the RESTRICTED TEXT TYPE element: ***Value 2.*** For [v1/2] metafiles, interpreter treatment of the 2 aspects of line cap shall be either: • in the style of one specific parameter value pair from the set of standard and registered values (excluding values 1) of the LINE CAP element. **No** Values = ? • in the style of any parameter value pair from the set of standard and registered values (excluding values 1) of the LINE CAP element. **Yes** For [v1/2] metafiles, interpreter treatment of the 2 aspects of edge cap shall be either: • in the style of one specific parameter value pair, from the set of standard and registered values (excluding values 1) of the EDGE CAP element. **No** Values = ? • in the style of any parameter value pair, from the set of standard and registered values (excluding	Drawing priority and mode: ***Priority shall correspond to the metafile order (i.e., primitives occurring later in the file shall overlay primitives occurring earliest in the file. Mode shall be "replacement" mode.)*** View surface clearing at picture start: ***Surface will be cleared upon the occurrence of BEGIN PICTURE BODY.*** Clipping: ***When CLIP INDICATOR is 'off', clipping shall be to the intersection of the device viewport and the device view surface limits. When CLIP INDICATOR is 'on', clipping shall be to the intersection of the clip rectangle, the VDC EXTENT, the device viewport, and the device view surface limits.*** Edge centering: ***Edges shall be centred on the ideal mathematically-defined edge of the area.*** Meaning of predefined line types and edge types: ***The exact on-off definitions for the predefined line types and edge types are not specified.*** Meaning of predefined hatch styles: ***The inter-line spacing is not specified. Use the latitudes of annex D.4.6.16 for the angular directions.*** In the absence of a LINE/MARKER/TEXT/EDGE CLIPPING MODE element, the interpreter treatment of LINE/MARKER/TEXT/EDGE CLIPPING MODE shall be: In the style of one specific parameter value, from the set of standardized values. **YES.** Specify which one: **SHAPE** In the style of any of the specific parameter values, from the set of standardized values. **NO** For [v1/v2] metafiles, text restriction method for RESTRICTED TEXT elements, chosen from the set of standard and registered styles of the RESTRICTED TEXT TYPE element: ***Value 2.*** For [v1/2] metafiles, interpreter treatment of the 2 aspects of line cap shall be either: • in the style of one specific parameter value pair from the set of standard and registered values (excluding values 1) of the LINE CAP element. **No** Values = ? • in the style of any parameter value pair from the set of standard and registered values (excluding values 1) of the LINE CAP element. **Yes** For [v1/2] metafiles, interpreter treatment of the 2 aspects of edge cap shall be either: • in the style of one specific parameter value pair, from the set of standard and registered values (excluding values 1) of the EDGE CAP element. **No** Values = ? • in the style of any parameter value pair, from the

	values 1) of the EDGE CAP element. **Yes** For [v1/2] metafiles, interpreter treatment of line join shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the LINE JOIN element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the LINE JOIN element. **Yes** For [v1/2] metafiles, interpreter treatment of edge join shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the EDGE JOIN element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the EDGE JOIN element. **Yes** For [v1/2] metafiles, interpreter treatment of line type continuation shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the LINE TYPE CONTINUATION element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the LINE TYPE CONTINUATION element. **Yes** For [v1/2] metafiles, interpreter treatment of edge type continuation shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the EDGE TYPE CONTINUATION element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the EDGE TYPE CONTINUATION element. **Yes** Other: ***None.***	set of standard and registered values (excluding values 1) of the EDGE CAP element. **Yes** For [v1/2] metafiles, interpreter treatment of line join shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the LINE JOIN element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the LINE JOIN element. **Yes** For [v1/2] metafiles, interpreter treatment of edge join shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the EDGE JOIN element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the EDGE JOIN element. **Yes** For [v1/2] metafiles, interpreter treatment of line type continuation shall be either: • **No**; in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the LINE TYPE CONTINUATION element. Value = ? • **Yes**; in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the LINE TYPE CONTINUATION element. For [v1/2] metafiles, interpreter treatment of edge type continuation shall be either: • in the style of one specific parameter value, from the set of standard and registered values (excluding value 1) of the EDGE TYPE CONTINUATION element. **No** Value = ? • in the style of any parameter value, from the set of standard and registered values (excluding value 1) of the EDGE TYPE CONTINUATION element. **Yes** Other: ***None.***
T.26.8	Same as Model Profile: **Yes**	
Error processing References: 9.5.7.6		Is error processing addressed? (yes/no) ***No.*** If yes, specify the action taken. Classification of error severity? Requirements for error recovery? Requirements for error reporting?

		Additional areas? Other: ***None.***
T.26.9	Same as Model Profile: **Yes**	
Reporting References: 9.5.7.7		Is reporting required? (yes/no) ***No.*** If yes, specify the action taken. Method and format of the reporting? Requirement to report any substitution, error, fallback behavior, mappings, or other behaviors? Additional areas? Other: ***None.***
T.26.10	Same as Model Profile: **Yes**	
Degeneracies References: 9.5.7.8 9.5.4.4 D.2 D.4		Is the interpretation of degenerate primitives addressed? (yes/no) ***Yes.*** If yes, for each primitive, specify the degeneracy including its source (i.e., intrinsic or computational). ***Intrinsically degenerate primitives shall be rendered as specified in annex D subsections: D.2.2, D.2.3, D.4.5.4 through D.4.5.8, D.4.5.11, and D.4.5.12. Interpreters are not required to detect computational degeneracy. If interpreters do detect computational degeneracies, they shall be rendered as specified in annex D subsections: D.2.2, D.2.3, D.4.5.4 through D.4.5.8, D.4.5.11, and D.4.5.12*** Other: ***None.***

Attachment 26.4

Colour requirements, Model Profile:

The colour mapping step (CMS) and colour rendering step (CRS) for each class of interpreters is as follows:

- monochrome:
  CMS
  all foreground information is mapped to one colour, background information to another colour.
  CRS
  all foreground information is mapped to one colour, background information to another colour.
- grayscale:
  CMS
  32 gray levels, the recommendations of annex D.3.2 is used to map colour to gray.
  CRS
  a unique representation of each of the levels of gray.
- full colour:
  CMS
  5R,9G,5B grid of RGB colour cube, plus a 32 gray levels (0-1), some of which are already on the grid.

CRS
a unique representation of all the "colours".

6.16 Binary Encoding Rules

6.16.1 Delimiter Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.12.1	Same as Model Profile: **Yes**	
NOOP [v1] References: Part 3, 8.2		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***None.*** Other: ***None.***

6.16.2 Metafile Descriptor Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.13.1	Same as Model Profile: **No**	
INTEGER PRECISION [v1] References: 7.3.4 Part 3, 8.3	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***16, 32*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***8, 16, or 32.*** Other: ***None.***
T.13.2	Same as Model Profile: **Yes**	
REAL PRECISION [v1] References: 7.3.5 Part 3, 8.3		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value?***(1, 16, 16) or (0, 9, 23).*** Other: ***None.***
T.13.3	Same as Model Profile: **No**	

INDEX PRECISION [v1] References: 7.3.6	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***16*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***8, 16, or 32.*** Other: ***None.***
T.13.4	Same as Model Profile: **Yes**	
COLOUR PRECISION [v1] References: 7.3.7		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***8 or 16.*** Other: ***None.***
T.13.5	Same as Model Profile: **Yes**	
COLOUR INDEX PRECISION [v1] References: 7.3.8		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***8 or 16.*** Other: ***None.***
T.13.6	Same as Model Profile: **No**	
NAME PRECISION [v2] References: 7.3.16 Part 3, 8.3	Element is: Required **No**; Permitted **No**; Prohibited **Yes** Any restrictions on the parameter value? ***8 or 16.*** Other: ***None.***	Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***16 or 32.*** Other: ***None.***

6.16.3 Control Elements

Element	Specifications - WebCGM 2.0 Profile	Specifications - Model Profile
T.14.1	Same as Model Profile: **Yes**	

VDC INTEGER PRECISION [v1] References: 7.5.1 Part 3, 8.5		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***16 or 32.*** Other: ***None.***
T.14.2	Same as Model Profile: **Yes**	
VDC REAL PRECISION [v1] References: 7.5.2 Part 3, 8.5		Element is: Required **No**; Permitted **Yes**; Any restrictions on the parameter value? ***(1, 16, 16) or (0, 9, 32).*** Other: ***None.***

6.17 Symbol Library

Note. The symbol library functionality of WebCGM 1.0 is removed from WebCGM 2.0.

6.18 Line and Edge Style Definitions

WebCGM supports both generic, but imprecise line types, and specific, precise line types. The realizations of line types 1..5 are described in general terms in the CGM standard (e.g., "dash-dot-dot"), and the realizations and constraints of the registered line types 6..15 are described in the ISO Register of Graphical Items (e.g., requirements for inking vertexes in certain engineering line types). Otherwise, the exact line patterns of implicit line types 1..15 are unconstrained. Where exact realizations of line types are expected and required, the LINE AND EDGE TYPE DEFINITION element should be used.

6.19 Hatch Style Definitions

WebCGM supports both generic, but imprecise hatch styles, and specific, precise hatch styles. The realizations of hatch styles 1..6 are described in general terms in the CGM standard. Otherwise, the exact hatch patterns of implicit hatch styles 1..6 are unconstrained. Where exact realizations of hatch styles are expected and required, the HATCH STYLE DEFINITION element should be used.

6.20 JPEG Compression within the Tile Element

This profile allows the use of JPEG restricted to the TILE element. The method is limited to BASELINE JPEG. BASELINE JPEG conforms to the process required for all DCT-based decoders. The colour selection mode of the TILE element shall always be direct, independent of the COLOUR SELECTION MODE in effect in the CGM. The cell colour precision parameter of the TILE shall always be 8-bit for BASELINE JPEG. The COLOUR model of the TILE element shall be defined in the method specific parameters element of the TILE. It can be the same or independent of the COLOUR MODEL of the CGM. BASELINE JPEG shall assume that the order of the spectral bands is the same order given by the colour model as defined by the method specific parameters. For example, if the model is RGB, each scan will compress the red component, followed by the green component, followed by the blue component. For the case where the colour model is "RGB related", the specific colour model shall be defined in the method specific parameters of the TILE element. The method specific parameters shall be present for each image compressed using BASELINE JPEG. The parameters shall be encoded as an SDR. The JPEG colour model parameter is required and is specified according to the rules of the INDEX PRECISION element. Valid values are:

- 0 - JPEG colour model is the same as the COLOUR MODEL of the CGM.
- 1 -1RGB
- 2 - CIELAB
- 3 - CIELUV

- 4 - CMYK
- 5 - RGB related

Values outside the range of 0-5 are not allowed. The JPEG colour submodel is required only when the JPEG colour model is "RGB related" and is specified according to the rules of the INDEX PRECISION ELEMENT. Valid values are:

- 0 - YCbCr
- 1 - YCrCb
- 2 - YUV
- 3 - YIQ
- 4 - YES
- 5 - ADT

Other values are not allowed.

Back to top of chapter

CGM Open specification - WebCGM 2.0 - Appendixes

Contents

A. Conformance

This section and its subsections are normative, unless otherwise indicated.

A.1 Conformance definitions

WebCGM 2.0 defines conformance for these classes of product:

- WebCGM 2.0 instances
- WebCGM viewers (static)
- WebCGM viewers (dynamic), including WebCGM DOM implementation
- XML Companion File (XCF) instances

WebCGM contains both static graphics functionality and dynamic behaviors functionality. Viewer conformance to the static graphics functionality can be measured for any kind of WebCGM viewer. Full viewer conformance to the dynamic behaviors specifications can only be measured in an environment of HTML-based documents and Web browsers. Therefore, full dynamic conformance of a viewer to all specifications in WebCGM 2.0 can only be measured for a WebCGM browser plugin (or equivalent architecture).

A.2 Validation tools

This subsection is informative.

One of the primary benefits of any CGM profile is the ability to insure interoperability through the use of validation tools against CGM instances and certification services for applications. Once an application has been certified through a testing service, behavior of that application is predictable under the constraints of the profile. Validation and certification tools and services which exist (or have existed) and can be leveraged for WebCGM are:

- Validation tools exist for metafile *instances*.
- A WebCGM 1.0 test suite (currently being upgraded for WebCGM 2.0).
- Previous test suites and certification services for *interpreters* for the closely related ATA profile -- several viewer, printer, and interpreter products that now also support WebCGM were certified for ATA compliance.

A.3 Obsolete and deprecated features

These features of WebCGM 1.0 have been made obsolete, and are not part of the WebCGM 2.0 standard:

- multiple pictures -- a valid WebCGM 2.0 instance may only contain one CGM picture; WebCGM 1.0 allowed multiple pictures.
- symbol libraries -- this 1.0 functionality was unused and unseen in the five years between WebCGM 1.0 and 2.0, therefore all elements associated with Symbol Libraries have been removed from WebCGM 2.0.
- continued Application Structures

These WebCGM 1.0 features are deprecated in WebCGM 2.0, and may be removed (made obsolete) in a future version:

- TILE compression types 0, 1, 2
- BITONAL TILE compression types 0, 1
- colour models sRGB and sRGB-alpha
- the 'view_context' object behavior in the URI fragment

A.4 Optional features

There are no optional features in WebCGM. Conforming static implementations must implement all static functionality as described herein. Conforming dynamic implementations

must implement all dynamic functionality and DOM functionality, as described herein.

B. References

B.1 Normative References

This section is normative.

All valid profiles of CGM conform to the ISO CGM standard. As are all profiles, the WebCGM profile is defined by reference to the ISO standard:

ISO/IEC 8632:1999(E)
Information technology - Computer graphics - Metafile for the storage and transfer of picture description information
- Part 1: Functional description
- Part 3: Binary encoding
- Part 4: Clear text encoding

Available at the ISO page of Publicly Available Standards. CGM:1999 was reaffirmed by ISO, without changes, at its 5-year review in 2004.

The other normative references of this profile are:

RFC-2396
Uniform Resource Identifiers (URI): Generic Syntax,
URL: *http://www.ietf.org/rfc/rfc2396.txt*

RFC-1951
Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC1951, Aladdin Enterprises, May 1996,
URL: *http://www.w3.org/Graphics/PNG/RFC-1951*

ISO/IEC 10646-1:1993, AM2:1996
Information technology - Universal multiple-octet coded character set (UCS)
- Part 1: Architecture and Basic Multilingual Plane
AMENDMENT 2: UCS Transformation Format 8 (UTF-8)

REC-png
PNG (Portable Network Graphics) Specification, Version 1.0, URL:
http://www.w3.org/TR/REC-png-multi

XML 1.0
XML 1.0, third edition, http://www.w3.org/TR/2004/REC-xml-20040204/.

B.2 Informative References

This section is informative (non-normative).

SVG 1.1
http://www.w3.org/TR/SVG11/

DOM Level 3
http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/

Xpointer
http://www.w3.org/TR/xptr/

Cascading Profiles
Definition and description of how to write a profile based on WebCGM as the starting point, for closely related technical application sectors. At http://www.cgmopen.org/technical/cascading-profiles.html

WebCGM 2.0 Requirements
The requirements used to define the new functionality for WebCGM 2.0. At http://www.cgmopen.org/technical/WebCGM_20_Requirements.html

B.3 Document Sources and Registration Authority

This section is informative (non-normative).

Copies of the ISO standards may be obtained from ISO:

ISO Central Secretariat
1, rue de Varembe
Case postale 56
CH-1211 Geneve 20
Switzerland

For the purpose of this Recommendation and according to the rules for the designation and operation of registration authorities in the ISO/IEC Directives, the ISO and IEC Councils have designated the following as the registration authority:

National Imagery and Mapping Agency (NIMA)
(Joint Interoperability Test Command)
The Registration Authority for Graphical Items
Building 57305
Fort Huachuca
Arizona
85613-7020

USA

For more information on the CGM standard itself, the CGM Open Web site has both bibliographic references, and short articles on CGM topics:

CGM Open
http://www.cgmopen.org/

The following World Wide Web sites have more information on CGM:

ISO/JTC1/SC24
http://www.cwi.nl/JTC1SC24/
W3C WebCGM Overview
http://www.w3.org/Graphics/WebCGM

C. What's new in WebCGM 2

This section is informative (non-normative).

1. DOM. A major functional addition is a limited WebCGM Document Object Model (DOM), carefully targetted at the WebCGM 2.0 requirements.
2. XCF. The second major functional addition is the definition of a standard XML Companion File for use by WebCGM (and related) applications, closely integrated with WebCGM DOM.
3. Event model. As an adjunct to the DOM definition, a more detailed event model has been specified, allowing definition and attachment of user event handlers to objects, and specifying how "pick" events are handled in a backward compatible way to WebCGM 1.0.
4. Single picture. Whereas ISO CGM:1999 allows multiple independent pictures in a metafile, and WebCGM 1.0 originally did, multiple pictures were deemed not amongst the requirements for WebCGM. To simplify WebCGM, they were deprecated in WebCGM 1.0 2nd Release, and are now removed from WebCGM 2.0.
5. Fragment syntax. Small changes to allowable values, in order to embody the single-picture rule, but the overall structure of the fragment is unchanged for backward compatibility.
6. Frag syntax and xcfterm. It is now possible in the link fragment to specify to load-and-apply an XML companion file before first display of the graphics of a targetted picture.
7. More object behaviors. The set of object behaviors that can be included in the link fragment syntax has be expanded to give much more user control.
8. Grnode. WebCGM 2.0 defines a purely graphical grouping mechanism, "graphical node", which groups graphical primitives as an Application Structure, but diallows the attributes

or properties that associate intelligence with objects.

9. New APS attributes. WebCGM defines new APS attributes of type 'visibility' and 'interactivity' that may be applied to most object (APS) types.
10. Obsolete features. A number of previously deprecated WebCGM 1.0 features are made obsolete (removed from WebCGM 2.0).
11. Deprecated features. A number WebCGM 1.0 features are deprecated in WebCGM 2.0 (may be removed in a future revision).
12. Symbol Libraries. This capability was in WebCGM 1.0, but was removed from WebCGM 2.0 because of lack of interest (POLYSYMBOL, several SYMBOL LIBRARY elements, some registered ESCAPEs, etc).
13. 2.0 and has added some graphical items that were deferred from WebCGM 1.0 (for expedience)
    - NUBS and NURBS
    - clipping to regions other than rectangles
    - increased sizes for Cell Array and Tile Array elements
    - Auxiliary Colour and Transparency
    - text-on-path (i.e., text layout along arbitrary paths)
    - increased sizes for Closed Figure elements

D. Glossary

(under construction, will be added to future revision)

Back to top of appendixes
