5.7 Fundamental Interfaces

The interfaces within this section are considered fundamental, and must be fully implemented by all conforming implementations WebCGM DOM. The WebCGM DOM presents WebCGM documents as a hierarchy of **WebCGMNode** objects that also implement more specialized interfaces. Some of the node types may have child nodes of various types, and others are leaf nodes that cann 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~~ enables the handling of ordered lists of WebCGMNodes. The **WebCGMEvent** interface provides contextual information regarding r 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.

No **Attributes**

5.7.3 Interface WebCGMMetafile

The `WebCGMMetafile` interface is the entry point to the entire WebCGM document. The interface exposes 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)~~ .which is the id parameter in the BEGIN METAFILE element in the CGM document.

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"| "grnode")	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.~~ Returns whether this node has an attribute with a given local name and namespace URI

No Parameters

Return Value

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

true when an attribute with a given local name and namespace URI is specified on this WebCGMNode, returns 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               setStyleAttrProperty(in WebCGMString style,
                                          in WebCGMString value);
  void               clearStyleAttrProoerty(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~~. empty WebCGMNodeList Only WebCGM App 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 Property

Set a style attribute at the picture level by name.

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

Style Property Name	WebCGMPicture level	APS level	Proberty 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 properties 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 cap 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 proberty value will increase or decrease the displayed character size by the value stated.

fill-color is the style properity 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 properity to modify.

value of type WebCGMString

The new value for the given style.

No Return value

No Exceptions

clearStyleProperity

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

Parameters

style of type WebCGMString

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

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         setStyleProperty(in WebCGMString style,
                                in WebCGMString value);
  void         clearStyleProperty(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

setStyleProperty

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

Parameters

style of type WebCGMString

The name of the style property to modify.

value of type WebCGMString

The new value for the given style.

No Return value

No Exceptions

clearStyleProperty

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

Parameters

style of type WebCGMString

The name of the style property. 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
