07 April 2004

# Appendix F: IDL Definitions

This appendix contains the complete OMG IDL [*OMG IDL*] for the Level 3 Document Object Model Core definitions.

The IDL files are also available as: http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/idl.zip

## dom.idl:

```
// File: dom.idl

#ifndef _DOM_IDL_
#define _DOM_IDL_

#pragma prefix "w3c.org"
module dom
{

  valuetype DOMString sequence<unsigned short>;

  typedef   unsigned long long DOMTimeStamp;

  typedef   any DOMUserData;

  typedef   Object DOMObject;

  interface DOMImplementation;
  interface DocumentType;
  interface Document;
  interface NodeList;
  interface NamedNodeMap;
  interface UserDataHandler;
  interface Element;
  interface TypeInfo;
  interface DOMLocator;

  exception DOMException {
    unsigned short   code;
  };
  // ExceptionCode
  const unsigned short      INDEX_SIZE_ERR                 = 1;
  const unsigned short      DOMSTRING_SIZE_ERR             = 2;
  const unsigned short      HIERARCHY_REQUEST_ERR          = 3;
  const unsigned short      WRONG_DOCUMENT_ERR             = 4;
  const unsigned short      INVALID_CHARACTER_ERR          = 5;
  const unsigned short      NO_DATA_ALLOWED_ERR            = 6;
  const unsigned short      NO_MODIFICATION_ALLOWED_ERR    = 7;
  const unsigned short      NOT_FOUND_ERR                  = 8;
  const unsigned short      NOT_SUPPORTED_ERR              = 9;
  const unsigned short      INUSE_ATTRIBUTE_ERR            = 10;
  // Introduced in DOM Level 2:
  const unsigned short      INVALID_STATE_ERR              = 11;
```

```
// Introduced in DOM Level 2:
const unsigned short      SYNTAX_ERR                     = 12;
// Introduced in DOM Level 2:
const unsigned short      INVALID_MODIFICATION_ERR       = 13;
// Introduced in DOM Level 2:
const unsigned short      NAMESPACE_ERR                  = 14;
// Introduced in DOM Level 2:
const unsigned short      INVALID_ACCESS_ERR             = 15;
// Introduced in DOM Level 3:
const unsigned short      VALIDATION_ERR                 = 16;
// Introduced in DOM Level 3:
const unsigned short      TYPE_MISMATCH_ERR              = 17;


// Introduced in DOM Level 3:
interface DOMStringList {
  DOMString          item(in unsigned long index);
  readonly attribute unsigned long   length;
  boolean            contains(in DOMString str);
};

// Introduced in DOM Level 3:
interface NameList {
  DOMString          getName(in unsigned long index);
  DOMString          getNamespaceURI(in unsigned long index);
  readonly attribute unsigned long   length;
  boolean            contains(in DOMString str);
  boolean            containsNS(in DOMString namespaceURI,
                                in DOMString name);
};

// Introduced in DOM Level 3:
interface DOMImplementationList {
  DOMImplementation  item(in unsigned long index);
  readonly attribute unsigned long   length;
};

// Introduced in DOM Level 3:
interface DOMImplementationSource {
  DOMImplementation  getDOMImplementation(in DOMString features);
  DOMImplementationList getDOMImplementationList(in DOMString features);
};

interface DOMImplementation {
  boolean            hasFeature(in DOMString feature,
                                in DOMString version);
  // Introduced in DOM Level 2:
  DocumentType       createDocumentType(in DOMString qualifiedName,
                                        in DOMString publicId,
                                        in DOMString systemId)
                                      raises(DOMException);
  // Introduced in DOM Level 2:
  Document           createDocument(in DOMString namespaceURI,
                                    in DOMString qualifiedName,
                                    in DocumentType doctype)
                                      raises(DOMException);
  // Introduced in DOM Level 3:
  DOMObject          getFeature(in DOMString feature,
                                in DOMString version);
};

interface Node {

  // NodeType
  const unsigned short      ELEMENT_NODE                   = 1;
  const unsigned short      ATTRIBUTE_NODE                 = 2;
  const unsigned short      TEXT_NODE                      = 3;
  const unsigned short      CDATA_SECTION_NODE             = 4;
  const unsigned short      ENTITY_REFERENCE_NODE          = 5;
  const unsigned short      ENTITY_NODE                    = 6;
  const unsigned short      PROCESSING_INSTRUCTION_NODE    = 7;
  const unsigned short      COMMENT_NODE                   = 8;
  const unsigned short      DOCUMENT_NODE                  = 9;
  const unsigned short      DOCUMENT_TYPE_NODE             = 10;
  const unsigned short      DOCUMENT_FRAGMENT_NODE         = 11;
  const unsigned short      NOTATION_NODE                  = 12;

  readonly attribute DOMString       nodeName;
           attribute DOMString       nodeValue;
                                        // raises(DOMException) on setting
                                        // raises(DOMException) on retrieval

  readonly attribute unsigned short  nodeType;
  readonly attribute Node            parentNode;
  readonly attribute NodeList        childNodes;
  readonly attribute Node            firstChild;
  readonly attribute Node            lastChild;
  readonly attribute Node            previousSibling;
  readonly attribute Node            nextSibling;
  readonly attribute NamedNodeMap    attributes;
  // Modified in DOM Level 2:
  readonly attribute Document        ownerDocument;
  // Modified in DOM Level 3:
  Node               insertBefore(in Node newChild,
                                  in Node refChild)
                                        raises(DOMException);
  // Modified in DOM Level 3:
  Node               replaceChild(in Node newChild,
                                  in Node oldChild)
                                        raises(DOMException);
  // Modified in DOM Level 3:
  Node               removeChild(in Node oldChild)
                                        raises(DOMException);
  // Modified in DOM Level 3:
  Node               appendChild(in Node newChild)
                                        raises(DOMException);
  boolean            hasChildNodes();
  Node               cloneNode(in boolean deep);
  // Modified in DOM Level 3:
  void               normalize();
  // Introduced in DOM Level 2:
  boolean            isSupported(in DOMString feature,
                                 in DOMString version);
  // Introduced in DOM Level 2:
  readonly attribute DOMString       namespaceURI;
  // Introduced in DOM Level 2:
           attribute DOMString       prefix;
                                        // raises(DOMException) on setting

  // Introduced in DOM Level 2:
  readonly attribute DOMString       localName;
  // Introduced in DOM Level 2:
  boolean            hasAttributes();
  // Introduced in DOM Level 3:
  readonly attribute DOMString       baseURI;

  // DocumentPosition
  const unsigned short      DOCUMENT_POSITION_DISCONNECTED = 0x01;
  const unsigned short      DOCUMENT_POSITION_PRECEDING    = 0x02;
  const unsigned short      DOCUMENT_POSITION_FOLLOWING    = 0x04;
  const unsigned short      DOCUMENT_POSITION_CONTAINS     = 0x08;
  const unsigned short      DOCUMENT_POSITION_CONTAINED_BY = 0x10;
  const unsigned short      DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC = 0x20;

  // Introduced in DOM Level 3:
  unsigned short     compareDocumentPosition(in Node other)
                                        raises(DOMException);
  // Introduced in DOM Level 3:
```

```
           attribute DOMString       textContent;
                                        // raises(DOMException) on setting
                                        // raises(DOMException) on retrieval

  // Introduced in DOM Level 3:
  boolean            isSameNode(in Node other);
  // Introduced in DOM Level 3:
  DOMString          lookupPrefix(in DOMString namespaceURI);
  // Introduced in DOM Level 3:
  boolean            isDefaultNamespace(in DOMString namespaceURI);
  // Introduced in DOM Level 3:
  DOMString          lookupNamespaceURI(in DOMString prefix);
  // Introduced in DOM Level 3:
  boolean            isEqualNode(in Node arg);
  // Introduced in DOM Level 3:
  DOMObject          getFeature(in DOMString feature,
                                in DOMString version);
  // Introduced in DOM Level 3:
  DOMUserData        setUserData(in DOMString key,
                                 in DOMUserData data,
                                 in UserDataHandler handler);
  // Introduced in DOM Level 3:
  DOMUserData        getUserData(in DOMString key);
};

interface NodeList {
  Node               item(in unsigned long index);
  readonly attribute unsigned long   length;
};

interface NamedNodeMap {
  Node               getNamedItem(in DOMString name);
  Node               setNamedItem(in Node arg)
                                        raises(DOMException);
  Node               removeNamedItem(in DOMString name)
                                        raises(DOMException);
  Node               item(in unsigned long index);
  readonly attribute unsigned long   length;
  // Introduced in DOM Level 2:
  Node               getNamedItemNS(in DOMString namespaceURI,
                                    in DOMString localName)
                                        raises(DOMException);
  // Introduced in DOM Level 2:
  Node               setNamedItemNS(in Node arg)
                                        raises(DOMException);
  // Introduced in DOM Level 2:
  Node               removeNamedItemNS(in DOMString namespaceURI,
                                       in DOMString localName)
                                        raises(DOMException);
};

interface CharacterData : Node {
           attribute DOMString       data;
                                        // raises(DOMException) on setting
                                        // raises(DOMException) on retrieval

  readonly attribute unsigned long   length;
  DOMString          substringData(in unsigned long offset,
                                   in unsigned long count)
                                        raises(DOMException);
  void               appendData(in DOMString arg)
                                        raises(DOMException);
  void               insertData(in unsigned long offset,
                                in DOMString arg)
                                        raises(DOMException);
  void               deleteData(in unsigned long offset,
                                in unsigned long count)
                                        raises(DOMException);
  void               replaceData(in unsigned long offset,
                                  in unsigned long count,
                                  in DOMString arg)
                                       raises(DOMException);
};

interface Attr : Node {
  readonly attribute DOMString       name;
  readonly attribute boolean         specified;
           attribute DOMString       value;
                                       // raises(DOMException) on setting

  // Introduced in DOM Level 2:
  readonly attribute Element         ownerElement;
  // Introduced in DOM Level 3:
  readonly attribute TypeInfo        schemaTypeInfo;
  // Introduced in DOM Level 3:
  readonly attribute boolean         isId;
};

interface Element : Node {
  readonly attribute DOMString       tagName;
  DOMString          getAttribute(in DOMString name);
  void               setAttribute(in DOMString name,
                                   in DOMString value)
                                       raises(DOMException);
  void               removeAttribute(in DOMString name)
                                       raises(DOMException);
  Attr               getAttributeNode(in DOMString name);
  Attr               setAttributeNode(in Attr newAttr)
                                       raises(DOMException);
  Attr               removeAttributeNode(in Attr oldAttr)
                                       raises(DOMException);
  NodeList           getElementsByTagName(in DOMString name);
  // Introduced in DOM Level 2:
  DOMString          getAttributeNS(in DOMString namespaceURI,
                                     in DOMString localName)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  void               setAttributeNS(in DOMString namespaceURI,
                                     in DOMString qualifiedName,
                                     in DOMString value)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  void               removeAttributeNS(in DOMString namespaceURI,
                                        in DOMString localName)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  Attr               getAttributeNodeNS(in DOMString namespaceURI,
                                         in DOMString localName)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  Attr               setAttributeNodeNS(in Attr newAttr)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  NodeList           getElementsByTagNameNS(in DOMString namespaceURI,
                                             in DOMString localName)
                                       raises(DOMException);
  // Introduced in DOM Level 2:
  boolean            hasAttribute(in DOMString name);
  // Introduced in DOM Level 2:
  boolean            hasAttributeNS(in DOMString namespaceURI,
                                     in DOMString localName)
                                       raises(DOMException);
  // Introduced in DOM Level 3:
  readonly attribute TypeInfo        schemaTypeInfo;
  // Introduced in DOM Level 3:
  void               setIdAttribute(in DOMString name,
                                     in boolean isId)
                                       raises(DOMException);
```

```
    // Introduced in DOM Level 3:
    void               setIdAttributeNS(in DOMString namespaceURI,
                                        in DOMString localName,
                                        in boolean isId)
                                        raises(DOMException);
    // Introduced in DOM Level 3:
    void               setIdAttributeNode(in Attr idAttr,
                                          in boolean isId)
                                        raises(DOMException);
  };

  interface Text : CharacterData {
    Text               splitText(in unsigned long offset)
                                        raises(DOMException);
    // Introduced in DOM Level 3:
    readonly attribute boolean         isElementContentWhitespace;
    // Introduced in DOM Level 3:
    readonly attribute DOMString       wholeText;
    // Introduced in DOM Level 3:
    Text               replaceWholeText(in DOMString content)
                                        raises(DOMException);
  };

  interface Comment : CharacterData {
  };

  // Introduced in DOM Level 3:
  interface TypeInfo {
    readonly attribute DOMString       typeName;
    readonly attribute DOMString       typeNamespace;

    // DerivationMethods
    const unsigned long       DERIVATION_RESTRICTION         = 0x00000001;
    const unsigned long       DERIVATION_EXTENSION           = 0x00000002;
    const unsigned long       DERIVATION_UNION               = 0x00000004;
    const unsigned long       DERIVATION_LIST                = 0x00000008;

    boolean            isDerivedFrom(in DOMString typeNamespaceArg,
                                     in DOMString typeNameArg,
                                     in unsigned long derivationMethod);
  };

  // Introduced in DOM Level 3:
  interface UserDataHandler {

    // OperationType
    const unsigned short      NODE_CLONED                    = 1;
    const unsigned short      NODE_IMPORTED                  = 2;
    const unsigned short      NODE_DELETED                   = 3;
    const unsigned short      NODE_RENAMED                   = 4;
    const unsigned short      NODE_ADOPTED                   = 5;

    void               handle(in unsigned short operation,
                              in DOMString key,
                              in DOMUserData data,
                              in Node src,
                              in Node dst);
  };

  // Introduced in DOM Level 3:
  interface DOMError {

    // ErrorSeverity
    const unsigned short      SEVERITY_WARNING               = 1;
    const unsigned short      SEVERITY_ERROR                 = 2;
    const unsigned short      SEVERITY_FATAL_ERROR           = 3;

    readonly attribute unsigned short  severity;
    readonly attribute DOMString       message;
    readonly attribute DOMString       type;
    readonly attribute DOMObject       relatedException;
    readonly attribute DOMObject       relatedData;
    readonly attribute DOMLocator      location;
  };

  // Introduced in DOM Level 3:
  interface DOMErrorHandler {
    boolean            handleError(in DOMError error);
  };

  // Introduced in DOM Level 3:
  interface DOMLocator {
    readonly attribute long            lineNumber;
    readonly attribute long            columnNumber;
    readonly attribute long            byteOffset;
    readonly attribute long            utf16Offset;
    readonly attribute Node            relatedNode;
    readonly attribute DOMString       uri;
  };

  // Introduced in DOM Level 3:
  interface DOMConfiguration {
    void               setParameter(in DOMString name, 
                                    in DOMUserData value)
                                        raises(DOMException);
    DOMUserData        getParameter(in DOMString name)
                                        raises(DOMException);
    boolean            canSetParameter(in DOMString name, 
                                       in DOMUserData value);
    readonly attribute DOMStringList   parameterNames;
  };

  interface CDATASection : Text {
  };

  interface DocumentType : Node {
    readonly attribute DOMString       name;
    readonly attribute NamedNodeMap    entities;
    readonly attribute NamedNodeMap    notations;
    // Introduced in DOM Level 2:
    readonly attribute DOMString       publicId;
    // Introduced in DOM Level 2:
    readonly attribute DOMString       systemId;
    // Introduced in DOM Level 2:
    readonly attribute DOMString       internalSubset;
  };

  interface Notation : Node {
    readonly attribute DOMString       publicId;
    readonly attribute DOMString       systemId;
  };

  interface Entity : Node {
    readonly attribute DOMString       publicId;
    readonly attribute DOMString       systemId;
    readonly attribute DOMString       notationName;
    // Introduced in DOM Level 3:
    readonly attribute DOMString       inputEncoding;
    // Introduced in DOM Level 3:
    readonly attribute DOMString       xmlEncoding;
    // Introduced in DOM Level 3:
    readonly attribute DOMString       xmlVersion;
  };

  interface EntityReference : Node {
  };

  interface ProcessingInstruction : Node {
    readonly attribute DOMString       target;
             attribute DOMString       data;
                                        // raises(DOMException) on setting

  };

  interface DocumentFragment : Node {
  };

  interface Document : Node {
    // Modified in DOM Level 3:
    readonly attribute DocumentType    doctype;
    readonly attribute DOMImplementation implementation;
    readonly attribute Element         documentElement;
    Element            createElement(in DOMString tagName)
                                        raises(DOMException);
    DocumentFragment   createDocumentFragment();
    Text               createTextNode(in DOMString data);
    Comment            createComment(in DOMString data);
    CDATASection       createCDATASection(in DOMString data)
                                        raises(DOMException);
    ProcessingInstruction createProcessingInstruction(in DOMString target,
                                                      in DOMString data)
                                        raises(DOMException);
    Attr               createAttribute(in DOMString name)
                                        raises(DOMException);
    EntityReference    createEntityReference(in DOMString name)
                                        raises(DOMException);
    NodeList           getElementsByTagName(in DOMString tagname);
    // Introduced in DOM Level 2:
    Node               importNode(in Node importedNode,
                                  in boolean deep)
                                        raises(DOMException);
    // Introduced in DOM Level 2:
    Element            createElementNS(in DOMString namespaceURI,
                                       in DOMString qualifiedName)
                                        raises(DOMException);
    // Introduced in DOM Level 2:
    Attr               createAttributeNS(in DOMString namespaceURI,
                                         in DOMString qualifiedName)
                                        raises(DOMException);
    // Introduced in DOM Level 2:
    NodeList           getElementsByTagNameNS(in DOMString namespaceURI,
                                              in DOMString localName);
    // Introduced in DOM Level 2:
    Element            getElementById(in DOMString elementId);
    // Introduced in DOM Level 3:
    readonly attribute DOMString       inputEncoding;
    // Introduced in DOM Level 3:
    readonly attribute DOMString       xmlEncoding;
    // Introduced in DOM Level 3:
             attribute boolean         xmlStandalone;
                                        // raises(DOMException) on setting

    // Introduced in DOM Level 3:
             attribute DOMString       xmlVersion;
                                        // raises(DOMException) on setting

    // Introduced in DOM Level 3:
             attribute boolean         strictErrorChecking;
    // Introduced in DOM Level 3:
             attribute DOMString       documentURI;
    // Introduced in DOM Level 3:
    Node               adoptNode(in Node source)
                                        raises(DOMException);
    // Introduced in DOM Level 3:
    readonly attribute DOMConfiguration domConfig;
    // Introduced in DOM Level 3:
    void               normalizeDocument();
```

```
    // Introduced in DOM Level 3:
    Node                  renameNode(in Node n,
                                     in DOMString namespaceURI,
                                     in DOMString qualifiedName)
                                        raises(DOMException);
  };
};

#endif // _DOM_IDL_
```

[previous] [next] [contents] [index]