previous next contents index

09 January 2003

Appendix B: IDL Definitions

This appendix contains the complete OMG IDL [OMG IDL] for the Level 2 Document Object Model HTML definitions.

Unfortunately the OMG IDL in this appendix is not conformant due to problems in the validator that was used to validate Level 1. The `readOnly` attribute on the `HTMLInputElement` and `HTMLTextAreaElement` interfaces, as well as the `object` attribute on the `HTMLAppletElement` interface, are not conformant with OMG IDL 2.2. The `valueType` attribute on the `HTMLParamElement` interface is not conformant with OMG IDL 2.3.1, which hadn't been released when DOM Level 1 [DOM Level 1] was published.

The IDL files are also available as: http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/idl.zip

html2.idl:

```
// File: html2.idl

#ifndef _HTML2_IDL_
#define _HTML2_IDL_

#include "dom.idl"

#pragma prefix "dom.w3c.org"
module html2
{

  typedef dom::DOMString DOMString;
  typedef dom::Node Node;
  typedef dom::Document Document;
  typedef dom::NodeList NodeList;
  typedef dom::Element Element;

  interface HTMLElement;
  interface HTMLFormElement;
  interface HTMLTableCaptionElement;
  interface HTMLTableSectionElement;

  interface HTMLCollection {
    readonly attribute unsigned long   length;
    Node               item(in unsigned long index);
    Node               namedItem(in DOMString name);
  };

  // Introduced in DOM Level 2:
  interface HTMLOptionsCollection {
             attribute unsigned long   length;
                                        // raises(dom::DOMException) on setting

    Node               item(in unsigned long index);
    Node               namedItem(in DOMString name);
  };
```

```
interface HTMLDocument : Document {
           attribute DOMString       title;
  readonly attribute DOMString       referrer;
  readonly attribute DOMString       domain;
  readonly attribute DOMString       URL;
           attribute HTMLElement     body;
  readonly attribute HTMLCollection  images;
  readonly attribute HTMLCollection  applets;
  readonly attribute HTMLCollection  links;
  readonly attribute HTMLCollection  forms;
  readonly attribute HTMLCollection  anchors;
           attribute DOMString       cookie;
                                      // raises(dom::DOMException) on setting

  void               open();
  void               close();
  void               write(in DOMString text);
  void               writeln(in DOMString text);
  NodeList           getElementsByName(in DOMString elementName);
};

interface HTMLElement : Element {
           attribute DOMString       id;
           attribute DOMString       title;
           attribute DOMString       lang;
           attribute DOMString       dir;
           attribute DOMString       className;
};

interface HTMLHtmlElement : HTMLElement {
           attribute DOMString       version;
};

interface HTMLHeadElement : HTMLElement {
           attribute DOMString       profile;
};

interface HTMLLinkElement : HTMLElement {
           attribute boolean         disabled;
           attribute DOMString       charset;
           attribute DOMString       href;
           attribute DOMString       hreflang;
           attribute DOMString       media;
           attribute DOMString       rel;
           attribute DOMString       rev;
           attribute DOMString       target;
           attribute DOMString       type;
};

interface HTMLTitleElement : HTMLElement {
           attribute DOMString       text;
};

interface HTMLMetaElement : HTMLElement {
           attribute DOMString       content;
           attribute DOMString       httpEquiv;
           attribute DOMString       name;
           attribute DOMString       scheme;
};

interface HTMLBaseElement : HTMLElement {
           attribute DOMString       href;
           attribute DOMString       target;
};

interface HTMLIsIndexElement : HTMLElement {
  readonly attribute HTMLFormElement form;
           attribute DOMString       prompt;
```

```
};

interface HTMLStyleElement : HTMLElement {
          attribute boolean         disabled;
          attribute DOMString       media;
          attribute DOMString       type;
};

interface HTMLBodyElement : HTMLElement {
          attribute DOMString       aLink;
          attribute DOMString       background;
          attribute DOMString       bgColor;
          attribute DOMString       link;
          attribute DOMString       text;
          attribute DOMString       vLink;
};

interface HTMLFormElement : HTMLElement {
  readonly attribute HTMLCollection  elements;
  readonly attribute long            length;
          attribute DOMString       name;
          attribute DOMString       acceptCharset;
          attribute DOMString       action;
          attribute DOMString       enctype;
          attribute DOMString       method;
          attribute DOMString       target;
  void               submit();
  void               reset();
};

interface HTMLSelectElement : HTMLElement {
  readonly attribute DOMString       type;
          attribute long            selectedIndex;
          attribute DOMString       value;
  // Modified in DOM Level 2:
          attribute unsigned long   length;
                                     // raises(dom::DOMException) on setting

  readonly attribute HTMLFormElement form;
  // Modified in DOM Level 2:
  readonly attribute HTMLOptionsCollection options;
          attribute boolean         disabled;
          attribute boolean         multiple;
          attribute DOMString       name;
          attribute long            size;
          attribute long            tabIndex;
  void               add(in HTMLElement element,
                         in HTMLElement before)
                                      raises(dom::DOMException);
  void               remove(in long index);
  void               blur();
  void               focus();
};

interface HTMLOptGroupElement : HTMLElement {
          attribute boolean         disabled;
          attribute DOMString       label;
};

interface HTMLOptionElement : HTMLElement {
  readonly attribute HTMLFormElement form;
  // Modified in DOM Level 2:
          attribute boolean         defaultSelected;
  readonly attribute DOMString       text;
  // Modified in DOM Level 2:
  readonly attribute long            index;
          attribute boolean         disabled;
          attribute DOMString       label;
          attribute boolean         selected;
```

```
           attribute DOMString       value;
};

interface HTMLInputElement : HTMLElement {
           attribute DOMString       defaultValue;
           attribute boolean         defaultChecked;
  readonly attribute HTMLFormElement form;
           attribute DOMString       accept;
           attribute DOMString       accessKey;
           attribute DOMString       align;
           attribute DOMString       alt;
           attribute boolean         checked;
           attribute boolean         disabled;
           attribute long            maxLength;
           attribute DOMString       name;
           attribute boolean         readOnly;
  // Modified in DOM Level 2:
           attribute unsigned long   size;
           attribute DOMString       src;
           attribute long            tabIndex;
  // Modified in DOM Level 2:
           attribute DOMString       type;
           attribute DOMString       useMap;
           attribute DOMString       value;
  void               blur();
  void               focus();
  void               select();
  void               click();
};

interface HTMLTextAreaElement : HTMLElement {
  // Modified in DOM Level 2:
           attribute DOMString       defaultValue;
  readonly attribute HTMLFormElement form;
           attribute DOMString       accessKey;
           attribute long            cols;
           attribute boolean         disabled;
           attribute DOMString       name;
           attribute boolean         readOnly;
           attribute long            rows;
           attribute long            tabIndex;
  readonly attribute DOMString       type;
           attribute DOMString       value;
  void               blur();
  void               focus();
  void               select();
};

interface HTMLButtonElement : HTMLElement {
  readonly attribute HTMLFormElement form;
           attribute DOMString       accessKey;
           attribute boolean         disabled;
           attribute DOMString       name;
           attribute long            tabIndex;
  readonly attribute DOMString       type;
           attribute DOMString       value;
};

interface HTMLLabelElement : HTMLElement {
  readonly attribute HTMLFormElement form;
           attribute DOMString       accessKey;
           attribute DOMString       htmlFor;
};

interface HTMLFieldSetElement : HTMLElement {
  readonly attribute HTMLFormElement form;
};

interface HTMLLegendElement : HTMLElement {
```

```
  readonly attribute HTMLFormElement form;
           attribute DOMString       accessKey;
           attribute DOMString       align;
};

interface HTMLUListElement : HTMLElement {
           attribute boolean         compact;
           attribute DOMString       type;
};

interface HTMLOListElement : HTMLElement {
           attribute boolean         compact;
           attribute long            start;
           attribute DOMString       type;
};

interface HTMLDListElement : HTMLElement {
           attribute boolean         compact;
};

interface HTMLDirectoryElement : HTMLElement {
           attribute boolean         compact;
};

interface HTMLMenuElement : HTMLElement {
           attribute boolean         compact;
};

interface HTMLLIElement : HTMLElement {
           attribute DOMString       type;
           attribute long            value;
};

interface HTMLDivElement : HTMLElement {
           attribute DOMString       align;
};

interface HTMLParagraphElement : HTMLElement {
           attribute DOMString       align;
};

interface HTMLHeadingElement : HTMLElement {
           attribute DOMString       align;
};

interface HTMLQuoteElement : HTMLElement {
           attribute DOMString       cite;
};

interface HTMLPreElement : HTMLElement {
           attribute long            width;
};

interface HTMLBRElement : HTMLElement {
           attribute DOMString       clear;
};

interface HTMLBaseFontElement : HTMLElement {
           attribute DOMString       color;
           attribute DOMString       face;
  // Modified in DOM Level 2:
           attribute long            size;
};

interface HTMLFontElement : HTMLElement {
           attribute DOMString       color;
           attribute DOMString       face;
           attribute DOMString       size;
};
```

```
interface HTMLHRElement : HTMLElement {
           attribute DOMString       align;
           attribute boolean         noShade;
           attribute DOMString       size;
           attribute DOMString       width;
};

interface HTMLModElement : HTMLElement {
           attribute DOMString       cite;
           attribute DOMString       dateTime;
};

interface HTMLAnchorElement : HTMLElement {
           attribute DOMString       accessKey;
           attribute DOMString       charset;
           attribute DOMString       coords;
           attribute DOMString       href;
           attribute DOMString       hreflang;
           attribute DOMString       name;
           attribute DOMString       rel;
           attribute DOMString       rev;
           attribute DOMString       shape;
           attribute long            tabIndex;
           attribute DOMString       target;
           attribute DOMString       type;
  void               blur();
  void               focus();
};

interface HTMLImageElement : HTMLElement {
           attribute DOMString       name;
           attribute DOMString       align;
           attribute DOMString       alt;
           attribute DOMString       border;
  // Modified in DOM Level 2:
           attribute long            height;
  // Modified in DOM Level 2:
           attribute long            hspace;
           attribute boolean         isMap;
           attribute DOMString       longDesc;
           attribute DOMString       src;
           attribute DOMString       useMap;
  // Modified in DOM Level 2:
           attribute long            vspace;
  // Modified in DOM Level 2:
           attribute long            width;
};

interface HTMLObjectElement : HTMLElement {
  readonly attribute HTMLFormElement form;
           attribute DOMString       code;
           attribute DOMString       align;
           attribute DOMString       archive;
           attribute DOMString       border;
           attribute DOMString       codeBase;
           attribute DOMString       codeType;
           attribute DOMString       data;
           attribute boolean         declare;
           attribute DOMString       height;
           attribute long            hspace;
           attribute DOMString       name;
           attribute DOMString       standby;
           attribute long            tabIndex;
           attribute DOMString       type;
           attribute DOMString       useMap;
           attribute long            vspace;
           attribute DOMString       width;
  // Introduced in DOM Level 2:
```

```
  readonly attribute Document        contentDocument;
};

interface HTMLParamElement : HTMLElement {
           attribute DOMString       name;
           attribute DOMString       type;
           attribute DOMString       value;
           attribute DOMString       valueType;
};

interface HTMLAppletElement : HTMLElement {
           attribute DOMString       align;
           attribute DOMString       alt;
           attribute DOMString       archive;
           attribute DOMString       code;
           attribute DOMString       codeBase;
           attribute DOMString       height;
  // Modified in DOM Level 2:
           attribute long            hspace;
           attribute DOMString       name;
  // Modified in DOM Level 2:
           attribute DOMString       object;
  // Modified in DOM Level 2:
           attribute long            vspace;
           attribute DOMString       width;
};

interface HTMLMapElement : HTMLElement {
  readonly attribute HTMLCollection  areas;
           attribute DOMString       name;
};

interface HTMLAreaElement : HTMLElement {
           attribute DOMString       accessKey;
           attribute DOMString       alt;
           attribute DOMString       coords;
           attribute DOMString       href;
           attribute boolean         noHref;
           attribute DOMString       shape;
           attribute long            tabIndex;
           attribute DOMString       target;
};

interface HTMLScriptElement : HTMLElement {
           attribute DOMString       text;
           attribute DOMString       htmlFor;
           attribute DOMString       event;
           attribute DOMString       charset;
           attribute boolean         defer;
           attribute DOMString       src;
           attribute DOMString       type;
};

interface HTMLTableElement : HTMLElement {
  // Modified in DOM Level 2:
           attribute HTMLTableCaptionElement caption;
                                     // raises(dom::DOMException) on setting

  // Modified in DOM Level 2:
           attribute HTMLTableSectionElement tHead;
                                     // raises(dom::DOMException) on setting

  // Modified in DOM Level 2:
           attribute HTMLTableSectionElement tFoot;
                                     // raises(dom::DOMException) on setting

  readonly attribute HTMLCollection  rows;
  readonly attribute HTMLCollection  tBodies;
           attribute DOMString       align;
```

```
           attribute DOMString       bgColor;
           attribute DOMString       border;
           attribute DOMString       cellPadding;
           attribute DOMString       cellSpacing;
           attribute DOMString       frame;
           attribute DOMString       rules;
           attribute DOMString       summary;
           attribute DOMString       width;
  HTMLElement        createTHead();
  void               deleteTHead();
  HTMLElement        createTFoot();
  void               deleteTFoot();
  HTMLElement        createCaption();
  void               deleteCaption();
  // Modified in DOM Level 2:
  HTMLElement        insertRow(in long index)
                                     raises(dom::DOMException);
  // Modified in DOM Level 2:
  void               deleteRow(in long index)
                                     raises(dom::DOMException);
};

interface HTMLTableCaptionElement : HTMLElement {
           attribute DOMString       align;
};

interface HTMLTableColElement : HTMLElement {
           attribute DOMString       align;
           attribute DOMString       ch;
           attribute DOMString       chOff;
           attribute long            span;
           attribute DOMString       vAlign;
           attribute DOMString       width;
};

interface HTMLTableSectionElement : HTMLElement {
           attribute DOMString       align;
           attribute DOMString       ch;
           attribute DOMString       chOff;
           attribute DOMString       vAlign;
  readonly attribute HTMLCollection  rows;
  // Modified in DOM Level 2:
  HTMLElement        insertRow(in long index)
                                     raises(dom::DOMException);
  // Modified in DOM Level 2:
  void               deleteRow(in long index)
                                     raises(dom::DOMException);
};

interface HTMLTableRowElement : HTMLElement {
  // Modified in DOM Level 2:
  readonly attribute long            rowIndex;
  // Modified in DOM Level 2:
  readonly attribute long            sectionRowIndex;
  // Modified in DOM Level 2:
  readonly attribute HTMLCollection  cells;
           attribute DOMString       align;
           attribute DOMString       bgColor;
           attribute DOMString       ch;
           attribute DOMString       chOff;
           attribute DOMString       vAlign;
  // Modified in DOM Level 2:
  HTMLElement        insertCell(in long index)
                                     raises(dom::DOMException);
  // Modified in DOM Level 2:
  void               deleteCell(in long index)
                                     raises(dom::DOMException);
};
```

```
    interface HTMLTableCellElement : HTMLElement {
      readonly attribute long            cellIndex;
               attribute DOMString       abbr;
               attribute DOMString       align;
               attribute DOMString       axis;
               attribute DOMString       bgColor;
               attribute DOMString       ch;
               attribute DOMString       chOff;
               attribute long            colSpan;
               attribute DOMString       headers;
               attribute DOMString       height;
               attribute boolean         noWrap;
               attribute long            rowSpan;
               attribute DOMString       scope;
               attribute DOMString       vAlign;
               attribute DOMString       width;
    };

    interface HTMLFrameSetElement : HTMLElement {
               attribute DOMString       cols;
               attribute DOMString       rows;
    };

    interface HTMLFrameElement : HTMLElement {
               attribute DOMString       frameBorder;
               attribute DOMString       longDesc;
               attribute DOMString       marginHeight;
               attribute DOMString       marginWidth;
               attribute DOMString       name;
               attribute boolean         noResize;
               attribute DOMString       scrolling;
               attribute DOMString       src;
      // Introduced in DOM Level 2:
      readonly attribute Document        contentDocument;
    };

    interface HTMLIFrameElement : HTMLElement {
               attribute DOMString       align;
               attribute DOMString       frameBorder;
               attribute DOMString       height;
               attribute DOMString       longDesc;
               attribute DOMString       marginHeight;
               attribute DOMString       marginWidth;
               attribute DOMString       name;
               attribute DOMString       scrolling;
               attribute DOMString       src;
               attribute DOMString       width;
      // Introduced in DOM Level 2:
      readonly attribute Document        contentDocument;
    };
  };

  #endif // _HTML2_IDL_
```

previous next contents index