Network Working Group
Request for Comments: 2070
Category: Standards Track

F. Yergeau
Alis Technologies
G. Nicol
Electronic Book Technologies
G. Adams
Spyglass
M. Duerst
University of Zurich

January 1997

Internationalization of the Hypertext Markup Language

Status of this Memo

Abstract

The Hypertext Markup Language (HTML) is a markup language used to create hypertext documents that are platform independent. Initially, the application of HTML on the World Wide Web was seriously restricted by its reliance on the ISO-8859-1 coded character set, which is appropriate only for Western European languages. Despite this restriction, HTML has been widely used with other languages, using other coded character sets or character encodings, at the expense of interoperability.

This document is meant to address the issue of the internationalization (i18n, i followed by 18 letters followed by n) of HTML by extending the specification of HTML and giving additional recommendations for proper internationalization support. A foremost consideration is to make sure that HTML remains a valid application of SGML, while enabling its use with all languages of the world.

Table of Contents

1. Introduction

The Hypertext Markup Language (HTML) is a markup language used to create hypertext documents that are platform independent. Initially, the application of HTML on the World Wide Web was seriously restricted by its reliance on the ISO-8859-1 coded character set, which is appropriate only for Western European languages. Despite this restriction, HTML has been widely used with other languages, using other coded character sets or character encodings, through various ad hoc extensions to the language [TAKADA].

This document is meant to address the issue of the internationalization of HTML by extending the specification of HTML and giving additional recommendations for proper internationalization support. It is in good part based on a paper by one of the authors on multilingualism on the WWW [NICOL]. A foremost consideration is to make sure that HTML remains a valid application of SGML, while enabling its use with all languages of the world.

The specific issues addressed are the SGML document character set to be used for HTML, the proper treatment of the charset parameter associated with the "text/html" content type and the specification of some additional elements and entities.

1.1 Scope

HTML has been in use by the World-Wide Web (WWW) global information initiative since 1990. This specification extends the capabilities of HTML 2.0 (RFC 1866), primarily by removing the restriction to the ISO-8859-1 coded character set [ISO-8859].

HTML is an application of ISO Standard 8879:1986, Information Processing Text and Office Systems -- Standard Generalized Markup Language (SGML) [ISO-8879]. The HTML Document Type Definition (DTD) is a formal definition of the HTML syntax in terms of SGML. This specification amends the DTD of HTML 2.0 in order to make it applicable to documents encompassing a character repertoire much larger than that of ISO-8859-1, while still remaining SGML conformant.

Both formal and actual development of HTML are advancing very fast. The features described in this document are designed so that they can (and should) be added to other forms of HTML besides that described in RFC 1866. Where indicated, attributes introduced here should be extended to the appropriate elements.

1.2 Conformance

This specification changes slightly the conformance requirements of HTML documents and HTML user agents.

1.2.1 Documents

All HTML 2.0 conforming documents remain conforming with this specification. However, the extensions introduced here make valid certain documents that would not be HTML 2.0 conforming, in particular those containing characters or character references outside of the repertoire of ISO 8859-1, and those containing markup introduced herein.

1.2.2. User agents

In addition to the requirements of RFC 1866, the following requirements are placed on HTML user agents.

To ensure interoperability and proper support for at least ISO-8859-1 in an environment where character encoding schemes other than ISO-8859-1 are present, user agents MUST correctly interpret the charset parameter accompanying an HTML document received from the network.

Furthermore, conforming user-agents MUST at least parse correctly all numeric character references within the range of ISO 10646-1 [ISO-10646].

Conforming user-agents are required to apply the BIDI presentation algorithm if they display right-to-left characters. If there is no displayable right-to-left character in a document, there is no need to apply BIDI processing.

2. The document character set

2.1. Reference processing model

This overview explains a reference processing model used for HTML, and in particular the SGML concept of a document character set. An actual implementation may widely differ in its internal workings from the model given below, but should behave as described to an outside observer.

Because there are various widely differing encodings of text, SGML does not directly address how the sequence of characters that constitutes an SGML document in the abstract sense are encoded by means of a sequence of octets (or occasionally bit groups of another length than 8) in a concrete realization of the document such as a computer file. This encoding is called the external character encoding of the concrete SGML document, and it should be carefully distinguished from the document character set of the abstract HTML document. SGML views the characters as a single set (called a "character repertoire"), and a "code set" that assigns an integer number (known as "character number") to each character in the repertoire. The document character set declaration defines what each of the character numbers represents [GOLD90, p. 451]. In most cases, an SGML DTD and all documents that refer to it have a single document character set, and all markup and data characters are part of this set.

HTML, as an application of SGML, does not directly address the question of the external character encoding. This is deferred to mechanisms external to HTML, such as MIME as used by the HTTP protocol or by electronic mail.

For the HTTP protocol [RFC2068], the external character encoding is indicated by the "charset" parameter of the "Content-Type" field of the header of an HTTP response. For example, to indicate that the transmitted document is encoded in the "JUNET" encoding of Japanese [RFC1468], the header will contain the following line:

Content-Type: text/html; charset=ISO-2022-JP

The term "charset" in MIME is used to designate a character encoding, rather than merely a coded character set as the term may suggest. A character encoding is a mapping (possibly many-to-one) of sequences of octets to sequences of characters taken from one or more character repertoires.

The HTTP protocol also defines a mechanism for the client to specify the character encodings it can accept. Clients and servers are

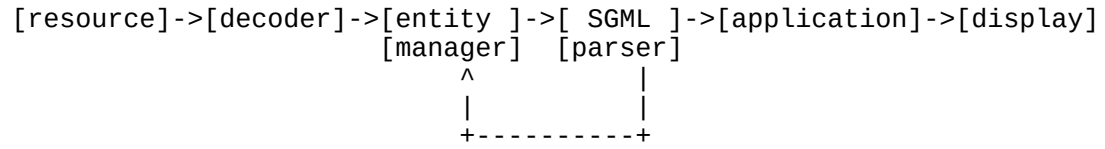
strongly requested to use these mechanisms to assure correct transmission and interpretation of any document. Provisions that can be taken to help correct interpretation, even in cases where a server or client do not yet use these mechanisms, are described in section 6.

Similarly, if HTML documents are transferred by electronic mail, the external character encoding is defined by the "charset" parameter of the "Content-Type" MIME header field [RFC2045], and defaults to US-ASCII in its absence.

No mechanisms are currently standardized for indicating the external character encoding of HTML documents transferred by FTP or accessed in distributed file systems.

In the case any other way of transferring and storing HTML documents are defined or become popular, it is advised that similar provisions be made to clearly identify the character encoding used and/or to use a single/default encoding capable of representing the widest range of characters used in an international context.

Whatever the external character encoding may be, the reference processing model translates it to the document character set specified in Section 2.2 before processing specific to SGML/HTML. The reference processing model can be depicted as follows:



The decoder is responsible for decoding the external representation of the resource to the document character set. The entity manager, the parser, and the application deal only with characters of the document character set. A display-oriented part of the application or the display machinery itself may again convert characters represented in the document character set to some other representation more suitable for their purpose. In any case, the entity manager, the parser, and the application, as far as character semantics are concerned, are using the HTML document character set only.

An actual implementation may choose, or not, to translate the document into some encoding of the document character set as described above; the behaviour described by this reference processing model can be achieved otherwise. This subject is well out of the scope of this specification, however, and the reader is invited to

consult the SGML standard [ISO-8879] or an SGML handbook [BRYAN88] [GOLD90] [VANH90] [SQ91] for further information.

The most important consequence of this reference processing model is that numeric character references are always resolved with respect to the fixed document character set, and thus to the same characters, whatever the external encoding actually used. For an example, see Section 2.2.

2.2. The document character set

The document character set, in the SGML sense, is the Universal Character Set (UCS) of ISO 10646:1993 [ISO-10646], as amended. Currently, this is code-by-code identical with the Unicode standard, version 1.1 [UNICODE].

> NOTE -- implementers should be aware that ISO 10646 is amended from time to time; 4 amendments have been adopted since the initial 1993 publication, none of which significantly affects this specification. A fifth amendment, now under consideration, will introduce incompatible changes to the standard: 6556 Korean Hangul syllables allocated between code positions 3400 and 4DFF (hexadecimal) will be moved to new positions (and 4516 new syllables added), thus making references to the old positions invalid. Since the Unicode consortium has already adopted a corresponding amendment for inclusion in the forthcoming Unicode 2.0, adoption of DAM 5 is considered likely and implementers should probably consider the old code positions as already invalid. Despite this one-time change, the relevant standard bodies have committed themselves not to change any allocated code position in the future. To encode Korean Hangul irrespective of these changes, the conjoining Hangul Jamo in the range 1110-11F9 can be used.

The adoption of this document character set implies a change in the SGML declaration specified in the HTML 2.0 specification (section 9.5 of [RFC1866]). The change amounts to removing the first BASESET specification and its accompanying DESCSET declaration, replacing them with the following declaration:

```
BASESET "ISO Registration Number 177//CHARSET
         ISO/IEC 10646-1:1993 UCS-4 with implementation level 3
         //ESC 2/5 2/15 4/6"
DESCSET  0   9     UNUSED
         9   2     9
         11  2     UNUSED
         13  1     13
         14  18    UNUSED
         32  95    32
         127 1     UNUSED
         128 32    UNUSED
         160 2147483486 160
```

Making the UCS the document character set does not create non-conformance of any expression, construct or document that is conforming to HTML 2.0. It does make conforming certain constructs that are not admissible in HTML 2.0. One consequence is that data characters outside the repertoire of ISO-8859-1, but within that of UCS-4 become valid SGML characters. Another is that the upper limit of the range of numeric character references is extended from 255 to 2147483645; thus, И is a valid reference to a "CYRILLIC CAPITAL LETTER I". [ERCS] is a good source of information on Unicode and SGML, although its scope and technical content differ greatly from this specification.

> NOTE -- the above SGML declaration, like that of HTML 2.0, specifies the character numbers 128 to 159 (80 to 9F hex) as UNUSED. This means that numeric character references within that range (e.g. ’) are illegal in HTML. Neither ISO 8859-1 nor ISO 10646 contain characters in that range, which is reserved for control characters.

Another change was made from the HTML 2.0 SGML declaration, in the belief that the latter did not express its authors' true intent. The syntax character set declaration was changed from ISO 646.IRV:1983 to the newer ISO 646.IRV:1991, the latter, but not the former, being identical with US-ASCII. In principle, this introduces an incompatibility with HTML 2.0, but in practice it should increase interoperability by i) having the SGML declaration say what everyone thinks and ii) making the syntax character set a proper subset of the document character set. The characters that differ between the two versions of ISO 646.IRV are not actually used to express HTML syntax.

ISO 10646-1:1993 is the most encompassing character set currently existing, and there is no other character set that could take its place as the document character set for HTML. If nevertheless for a specific application there is a need to use characters outside this standard, this should be done by avoiding any conflicts with present

or future versions of ISO 10646, i.e. by assigning these characters to a private zone of the UCS-4 coding space [ISO-10646 section 11]. Also, it should be borne in mind that such a use will be highly unportable; in many cases, it may be better to use inline bitmaps.

2.3. Undisplayable characters

With the document character set being the full ISO 10646, the possibility that a character cannot be displayed due to lack of appropriate resources (fonts) cannot be avoided. Because there are many different things that can be done in such a case, this document does not prescribe any specific behaviour. Depending on the implementation, this may also be handled by the underlaying display system and not the application itself. The following considerations, however, may be of help:

- A clearly visible, but unobtrusive behaviour should be preferred. Some documents may contain many characters that cannot be rendered, and so showing an alert for each of them is not the right thing to do.

- In case a numeric representation of the missing character is given, its hexadecimal (not decimal) form is to be preferred, because this form is used in character set standards [ERCS].

3. The LANG attribute

Language tags can be used to control rendering of a marked up document in various ways: glyph disambiguation, in cases where the character encoding is not sufficient to resolve to a specific glyph; quotation marks; hyphenation; ligatures; spacing; voice synthesis; etc. Independently of rendering issues, language markup is useful as content markup for purposes such as classification and searching.

Since any text can logically be assigned a language, almost all HTML elements admit the LANG attribute. The DTD reflects this; the only elements in this version of HTML without the LANG attribute are BR, HR, BASE, NEXTID, and META. It is also intended that any new element introduced in later versions of HTML will admit the LANG attribute, unless there is a good reason not to do so.

The language attribute, LANG, takes as its value a language tag that identifies a natural language spoken, written, or otherwise conveyed by human beings for communication of information to other human beings. Computer languages are explicitly excluded.

The syntax and registry of HTML language tags is the same as that defined by RFC 1766 [RFC1766]. In summary, a language tag is composed of one or more parts: A primary language tag and a possibly empty series of subtags:

```
language-tag  = primary-tag *( "-" subtag )
primary-tag   = 1*8ALPHA
subtag        = 1*8ALPHA
```

Whitespace is not allowed within the tag and all tags are case-insensitive. The namespace of language tags is administered by the IANA. Example tags include:

```
en, en-US, en-cockney, i-cherokee, x-pig-latin
```

In the context of HTML, a language tag is not to be interpreted as a single token, as per RFC 1766, but as a hierarchy. For example, a user agent that adjusts rendering according to language should consider that it has a match when a language tag in a style sheet entry matches the initial portion of the language tag of an element. An exact match should be preferred. This interpretation allows an element marked up as, for instance, "en-US" to trigger styles corresponding to, in order of preference, US-English ("en-US") or 'plain' or 'international' English ("en").

NOTE -- using the language tag as a hierarchy does not imply that all languages with a common prefix will be understood by those fluent in one or more of those languages; it simply allows the user to request this commonality when it is true for that user.

The rendering of elements may be affected by the LANG attribute. For any element, the value of the LANG attribute overrides the value specified by the LANG attribute of any enclosing element and the value (if any) of the HTTP Content-Language header. If none of these are set, a suitable default, perhaps controlled by user preferences, by automatic context analysis or by the user's locale, should be used to control rendering.

4. Additional entities, attributes and elements

4.1. Full Latin-1 entity set

According to the suggestion of section 14 of [RFC1866], the set of Latin-1 entities is extended to cover the whole right part of ISO-8859-1 (all code positions with the high-order bit set), including the already commonly used , © and ®. The names of the entities are taken from the appendices of SGML [ISO-8879]. A list is provided in section 7.3 of this specification.

4.2. Markup for language-dependent presentation

4.2.1. Overview

For the correct presentation of text in certain languages (irrespective of formatting issues), some support in the form of additional entities and elements is needed.

In particular, the following features are dealt with:

- Markup of bidirectional text, i.e. text where left-to-right and right-to-left scripts are mixed.

- Control of cursive joining behaviour in contexts where the default behaviour is not appropriate.

- Language-dependent rendering of short (in-line) quotations.

- Better justification control for languages where this is important.

- Superscripts and subscripts for languages where they appear as part of general text.

Some of the above features need very little additional support; others need more. The additional features are introduced below with brief comments only. Explanations on cursive joining behaviour and bidirectional text follow later. For cursive joining behaviour and bidirectional text, this document follows [UNICODE] in that: i) character semantics, where applicable, are identical to [UNICODE], and ii) where functionality is moved to HTML as a higher level protocol, this is done in a way that allows straightforward conversion to the lower-level mechanisms defined in [UNICODE].

4.2.2. List of entities, elements, and attributes

First, a generic container is needed to carry the LANG and DIR (see below) attributes in cases where no other element is appropriate; the SPAN element is introduced for that purpose.

A set of named character entities is added for use with bidirectional rendering and cursive joining control:

```
<!ENTITY zwnj CDATA "&#8204;"--=zero width non-joiner-->
<!ENTITY zwj  CDATA "&#8205;"--=zero width joiner-->
<!ENTITY lrm  CDATA "&#8206;"--=left-to-right mark-->
<!ENTITY rlm  CDATA "&#8207;"--=right-to-left mark-->
```

These entities can be used in place of the corresponding formatting characters whenever convenient, for example to ease keyboard entry or when a formatting character is not available in the character encoding of the document.

Next, an attribute called DIR is introduced, restricted to the values LTR (left-to-right) and RTL (right-to-left), for the indication of directionality in the context of bidirectional text (see 4.2.4 below for details). Since any text and many other elements (e.g. tables) can logically be assigned a directionality, all elements except BR, HR, BASE, NEXTID, and META admit this attribute. The DTD reflects this. It is also intended that any new element introduced in later versions of HTML will admit the DIR attribute, unless there is a good reason not to do so.

A new phrase-level element called BDO (BIDI Override) is introduced, which requires the DIR attribute to specify whether the override is left-to-right or right-to-left. This element is required for bidirectional text control; for detailed explanations, see section 4.2.4.

The phrase-level element Q is introduced to allow language-dependent rendering of short quotations depending on language and platform capability. As the following examples show (rather poorly, because of the character set restriction of Internet specifications), the quotation marks surrounding the quotation are particularly affected: "a quotation in English", ‘another, slightly better one’, ,,a quotation in German’’, << a quotation in French >>. The contents of the Q element does not include quotation marks, which have to be added by the rendering process.

> NOTE -- Q elements can be nested. Many languages use different quotation styles for outer and inner quotations, and this should be respected by user-agents implementing this element.

NOTE -- minimal support for the Q element is to surround the contents with some kind of quotes, like the plain ASCII double quotes. As this is rather easy to implement, and as the lack of any visible quotes may affect the perceived meaning of the text, user-agent implementors are strongly requested to provide at least this minimal level of support.

Many languages require superscript text for proper rendering: as an example, the French "Mlle Dupont" should have "lle" in superscript. The SUP element, and its sibling SUB for subscript text, are introduced to allow proper markup of such text. SUP and SUB contents are restricted to PCDATA to avoid nesting problems.

Finally, in many languages text justification is much more important than it is in Western languages, and justifies markup. The ALIGN attribute, admitting values of LEFT, RIGHT, CENTER and JUSTIFY, is added to a selection of elements where it makes sense (the block-like P, HR, H1 to H6, OL, UL, DIR, MENU, LI, BLOCKQUOTE and ADDRESS). If a user-agent chooses to have LEFT as a default for blocks of left-to-right directionality, it should use RIGHT for blocks of right-to-left directionality.

NOTE -- RFC 1866 section 4.2.2 specifies that an HTML user agent should treat an end of line as a word space, except in preformatted text. This should be interpreted in the context of the script being processed, as the way words are separated in writing is script-dependent. For some scripts (e.g. Latin), a word space is just a space, but in other scripts (e.g. Thai) it is a zero-width word separator, whereas in yet other scripts (e.g. Japanese) it is nothing at all, i.e. totally ignored.

NOTE -- the SOFT HYPHEN character (U+00AD) needs special attention from user-agent implementers. It is present in many character sets (including the whole ISO 8859 series and, of course, ISO 10646), and can always be included by means of the reference . Its semantics are different from the plain HYPHEN: it indicates a point in a word where a line break is allowed. If the line is indeed broken there, a hyphen must be displayed at the end of the first line. If not, the character is not dispalyed at all. In operations like searching and sorting, it must always be ignored.

In the DTD, the LANG and DIR attributes are grouped together in a parameter entity called attrs. To parallel RFC 1942 [RFC1942], the ID and CLASS attributes are also included in attrs. The ID and CLASS attributes are required for use with style sheets, and RFC 1942 defines them as follows:

ID Used to define a document-wide identifier. This can be used for naming positions within documents as the destination of a hypertext link. It may also be used by style sheets for rendering an element in a unique style. An ID attribute value is an SGML NAME token. NAME tokens are formed by an initial letter followed by letters, digits, "-" and "." characters. The letters are restricted to A-Z and a-z.

CLASS A space separated list of SGML NAME tokens. CLASS names specify that the element belongs to the corresponding named classes. It allows authors to distinguish different roles played by the same tag. The classes may be used by style sheets to provide different renderings as appropriate to these roles.

4.2.3. Cursive joining behaviour

Markup is needed in some cases to force cursive joining behavior in contexts in which it would not normally occur, or to block it when it would normally occur.

The zero-width joiner and non-joiner (‍ and ‌) are used to control cursive joining behaviour. For example, ARABIC LETTER HEH is used in isolation to abbreviate "Hijri" (the Islamic calendrical system); however, the initial form of the letter is desired, because the isolated form of HEH looks like the digit five as employed in Arabic script. This is obtained by following the HEH with a zero-width joiner whose only effect is to provide context. In Persian texts, there are cases where a letter that normally would join a subsequent letter in a cursive connection does not. Here a zero-width non- joiner is used.

4.2.4. Bidirectional text

Many languages are written in horizontal lines from left to right, while others are written from right to left. When both writing directions are present, one talks of bidirectional text (BIDI for short). BIDI text requires markup in special circumstances where ambiguities as to the directionality of some characters have to be resolved. This markup affects the ability to render BIDI text in a semantically legible fashion. That is, without this special BIDI markup, cases arise which would prevent *any* rendering whatsoever

that reflected the basic meaning of the text. Plain text may contain BIDI markup in the form of special-purpose formatting characters.

This is also possible in HTML, which includes the five BIDI-related formatting characters (202A - 202E) of ISO 10646. As an alternative, HTML provides equivalent SGML markup.

BIDI is a complex issue, and conversion of logical text sequences to display sequences has to be done according to the algorithm and character properties specified in [UNICODE]. Here, explanations are given only as far as they are needed to understand the necessity of the features introduced and to define their exact semantics.

The Unicode BIDI algorithm is based on the individual characters of a text being stored in logical order, that is the order in which they are normally input and in which the corresponding sounds are normally spoken. To make rendering of logical order text possible, the algorithm assigns a directionality property to each character, e.g. Latin letters are specified to have a left-to-right direction, Arabic and Hebrew characters have a right-to-left direction.

The left-to-right and right-to-left marks (‎ and ‏) are used to disambiguate directionality of neutral characters. For example, when a double quote sits between an Arabic and a Latin letter, its direction is ambiguous; if a directional mark is added on one side such that the quotation mark is surrounded by characters of only one directionality, the ambiguity is removed. These characters are like zero width spaces which have a directional property (but no word/line break property).

Nested embeddings of contra-directional text runs, due to nested quotations or to the pasting of text from one BIDI context to another, is also a case where the implicit directionality of characters is not sufficient, requiring markup. Also, it is frequently desirable to specify the basic directionality of a block of text. For these purposes, the DIR attribute is used.

On block-type elements, the DIR attribute indicates the base directionality of the text in the block; if omitted it is inherited from the parent element. The default directionality of the overall HTML document is left-to-right.

On inline elements, it makes the element start a new embedding level (to be explained below); if omitted the inline element does not start a new embedding level.

NOTE -- the PRE, XMP and LISTING elements admit the DIR attribute. Their contents should not be considered as preformatted with respect to bidirectional layout, but the BIDI algorithm should be applied to each line of text.

Following is an example of a case where embedding is needed, showing its effect:

Given the following latin (upper case) and arabic (lower case) letters in backing store with the specified embeddings:

<SPAN DIR=LTR> AB <SPAN DIR=RTL> xy <SPAN DIR=LTR> CD </SPAN> zw </SPAN> EF </SPAN>

One gets the following rendering (with [] showing the directional transitions):

[AB [wz [CD] yx] EF]

On the other hand, without this markup and with a base direction of LTR one gets the following rendering:

[AB [yx] CD [wz] EF]

Notice that yx is on the left and wz on the right unlike the above case where the embedding levels are used. Without the embedding markup one has at most two levels: a base directional level and a single counterflow directional level.

The DIR attribute on inline elements is equivalent to the formatting characters LEFT-TO-RIGHT EMBEDDING (202A) and RIGHT-TO-LEFT EMBEDDING (202B) of ISO 10646. The end tag of the element is equivalent to the POP DIRECTIONAL FORMATTING (202C) character.

Directional override, as provided by the BDO element, is needed to deal with unusual short pieces of text in which directionality cannot be resolved from context in an unambiguous fashion. For example, it can be used to force left-to-right (or right-to-left) display of part numbers composed of Latin letters, digits and Hebrew letters.

The effect of BDO is to force the directionality of all characters within it to the value of DIR, irrespective of their intrinsic directional properties. It is equivalent to using the LEFT-TO-RIGHT OVERRIDE (202D) or RIGHT-TO-LEFT OVERRIDE (202E) characters of ISO 10646, the end tag again being equivalent to the POP DIRECTIONAL FORMATTING (202C) character.

NOTE -- authors and authoring software writers should be aware that conflicts can arise if the DIR attribute is used on inline elements (including BDO) concurrently with the use of the corresponding ISO 10646 formatting characters.

Preferably one or the other should be used exclusively; the markup method is better able to guarantee document structural integrity, and alleviates some problems when editing bidirectional HTML text with a simple text editor, but some software may be more apt at using the 10646 characters. If both methods are used, great care should be exercised to insure proper nesting of markup and directional embedding or override; otherwise, rendering results are undefined.

5. Forms

5.1. DTD additions

It is natural to expect input in any language in forms, as they provide one of the only ways of obtaining user input. While this is primarily a UI issue, there are some things that should be specified at the HTML level to guide behavior and promote interoperability.

To ensure full interoperability, it is necessary for the user agent (and the user) to have an indication of the character encoding(s) that the server providing a form will be able to handle upon submission of the filled-in form. Such an indication is provided by the ACCEPT-CHARSET attribute of the INPUT and TEXTAREA elements, modeled on the HTTP Accept-Charset header (see [HTTP-1.1]), which contains a space and/or comma delimited list of character sets acceptable to the server. A user agent may want to somehow advise the user of the contents of this attribute, or to restrict his possibility to enter characters outside the repertoires of the listed character sets.

NOTE -- The list of character sets is to be interpreted as an EXCLUSIVE-OR list; the server announces that it is ready to accept any ONE of these character encoding schemes for each part of a multipart entity. The client may perform character encoding translation to satisfy the server if necessary.

NOTE -- The default value for the ACCEPT-CHARSET attribute of an INPUT or TEXTAREA element is the reserved value "UNKNOWN". A user agent may interpret that value as the character encoding scheme that was used to transmit the document containing that element.

5.2. Form submission

The HTML 2.0 form submission mechanism, based on the "application/x-www-form-urlencoded" media type, is ill-equipped with regard to internationalization. In fact, since URLs are restricted to ASCII characters, the mechanism is akward even for ISO-8859-1 text. Section 2.2 of [RFC1738] specifies that octets may be encoded using the "%HH" notation, but text submitted from a form is composed of characters, not octets. Lacking a specification of a character encoding scheme, the "%HH" notation has no well-defined meaning.

The best solution is to use the "multipart/form-data" media type described in [RFC1867] with the POST method of form submission. This mechanism encapsulates the value part of each name-value pair in a body-part of a multipart MIME body that is sent as the HTTP entity; each body part can be labeled with an appropriate Content-Type, including if necessary a charset parameter that specifies the character encoding scheme. The changes to the DTD necessary to support this method of form submission have been incorporated in the DTD included in this specification.

A less satisfactory solution is to add a MIME charset parameter to the "application/x-www-form-urlencoded" media type specifier sent along with a POST method form submission, with the understanding that the URL encoding of [RFC1738] is applied on top of the specified character encoding, as a kind of implicit Content-Transfer-Encoding.

One problem with both solutions above is that current browsers do not generally allow for bookmarks to specify the POST method; this should be improved. Conversely, the GET method could be used with the form data transmitted in the body instead of in the URL. Nothing in the protocol seems to prevent it, but no implementations appear to exist at present.

How the user agent determines the encoding of the text entered by the user is outside the scope of this specification.

> NOTE -- Designers of forms and their handling scripts should be aware of an important caveat: when the default value of a field (the VALUE attribute) is returned upon form submission (i.e. the user did not modify this value), it cannot be guaranteed to be transmitted as a sequence of octets identical to that in the source document -- only as a possibly different but valid encoding of the same sequence of text elements. This may be true even if the encoding of the document containing the form and that used for submission are the same.

Differences can occur when a sequence of characters can be represented by various sequences of octets, and also when a composite sequence (a base character plus one or more combining diacritics) can be represented by either a different but equivalent composite sequence or by a fully precomposed character. For instance, the UCS-2 sequence 00EA+0323 (LATIN SMALL LETTER E WITH CIRCUMFLEX ACCENT + COMBINING DOT BELOW) may be transformed into 1EC7 (LATIN SMALL LETTER E WITH CIRCUMFLEX ACCENT AND DOT BELOW), into 0065+0302+0323 (LATIN SMALL LETTER E + COMBINING CIRCUMFLEX ACCENT + COMBINING DOT BELOW), as well as into other equivalent composite sequences.

6. External character encoding issues

Proper interpretation of a text document requires that the character encoding scheme be known. Current HTTP servers, however, do not generally include an appropriate charset parameter with the Content-Type header. This is bad behaviour, which is even encouraged by the continued existence of browsers that declare an unrecognized media type when they receive a charset parameter. User agent implementators are strongly encouraged to make their software tolerant of this parameter, even if they cannot take advantage of it. Proper labelling is highly desirable, but some preventive measures can be taken to minimize the detrimental effects of its absence:

In the case where a document is accessed from a hyperlink in an origin HTML document, a CHARSET attribute is added to the attribute list of elements with link semantics (A and LINK), specifically by adding it to the linkExtraAttributes entity. The value of that attribute is to be considered a hint to the User Agent as to the character encoding scheme used by the resource pointed to by the hyperlink; it should be the appropriate value of the MIME charset parameter for that resource.

In any document, it is possible to include an indication of the encoding scheme like the following, as early as possible within the HEAD of the document:

```
<META HTTP-EQUIV="Content-Type"
 CONTENT="text/html; charset=ISO-2022-JP">
```

This is not foolproof, but will work if the encoding scheme is such that ASCII-valued octets stand for ASCII characters only at least until the META element is parsed. Note that there are better ways for a server to obtain character encoding information, instead of the unreliable META above; see [NICOL2] for some details and a proposal.

For definiteness, the "charset" parameter received from the source of the document should be considered the most authoritative, followed in order of preference by the contents of a META element such as the above, and finally the CHARSET parameter of the anchor that was followed (if any).

When HTML text is transmitted directly in UCS-2 or UCS-4 form, the question of byte order arises: does the high-order byte of each multi-byte character come first or last? For definiteness, this specification recommends that UCS-2 and UCS-4 be transmitted in big-endian byte order (high order byte first), which corresponds to the established network byte order for two- and four-byte quantities, to the ISO 10646 requirement and Unicode recommendation for serialized text data and to RFC 1641. Furthermore, to maximize chances of proper interpretation, it is recommended that documents transmitted as UCS-2 or UCS-4 always begin with a ZERO-WIDTH NON-BREAKING SPACE character (hexadecimal FEFF or 0000FEFF) which, when byte-reversed becomes number FFFE or FFFE0000, a character guaranteed to be never assigned. Thus, a user-agent receiving an FFFE as the first octets of a text would know that bytes have to be reversed for the remainder of the text.

There exist so-called UCS Transformation Formats than can be used to transmit UCS data, in addition to UCS-2 and UCS-4. UTF-7 [RFC1642] and UTF-8 [UTF-8] have favorable properties (no byte-ordering problem, different flavours of ASCII compatibility) that make them worthy of consideration, especially for transmission of multilingual text. Another encoding scheme, MNEM [RFC1345], also has interesting properties and the capability to transmit the full UCS. The UTF-1 transformation format of ISO 10646:1993 (registered by IANA as ISO-10646-UTF-1), has been removed from ISO 10646 by amendment 4, and should not be used.

7. HTML Public Text

7.1. HTML DTD

This section contains a DTD for HTML based on the HTML 2.0 DTD of RFC 1866, incorporating the changes for file upload as specified in RFC 1867, and the changes deriving from this document.

```
<!--    html.dtd

        Document Type Definition for the HyperText Markup Language,
        extended for internationalisation (HTML DTD)

        Last revised: 96/08/07

     Authors: Daniel W. Connolly <connolly@w3.org>
              Francois Yergeau <yergeau@alis.com>
     See Also:
       http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html
-->

<!ENTITY % HTML.Version
        "-//IETF//DTD HTML i18n//EN"

        -- Typical usage:

            <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML i18n//EN">
            <html>
            ...
            </html>
        --
        >


<!--============ Feature Test Entities ========================-->

<!ENTITY % HTML.Recommended "IGNORE"
     -- Certain features of the language are necessary for
        compatibility with widespread usage, but they may
        compromise the structural integrity of a document.
        This feature test entity enables a more prescriptive
        document type definition that eliminates
        those features.
     -->

<![ %HTML.Recommended [
        <!ENTITY % HTML.Deprecated "IGNORE">
]]>
```

```
   <!ENTITY % HTML.Deprecated "INCLUDE"
        -- Certain features of the language are necessary for
           compatibility with earlier versions of the specification,
           but they tend to be used and implemented inconsistently,
           and their use is deprecated. This feature test entity
           enables a document type definition that eliminates
           these features.
        -->

   <!ENTITY % HTML.Highlighting "INCLUDE"
        -- Use this feature test entity to validate that a
           document uses no highlighting tags, which may be
           ignored on minimal implementations.
        -->

   <!ENTITY % HTML.Forms "INCLUDE"
           -- Use this feature test entity to validate that a document
              contains no forms, which may not be supported in minimal
              implementations
           -->

   <!--============== Imported Names ==============================-->

   <!ENTITY % Content-Type "CDATA"
           -- meaning an internet media type
              (aka MIME content type, as per RFC2045)
           -->

   <!ENTITY % HTTP-Method "GET | POST"
           -- as per HTTP specification, RFC2068
           -->

   <!--========= DTD "Macros" =====================-->

   <!ENTITY % heading "H1|H2|H3|H4|H5|H6">

   <!ENTITY % list " UL | OL | DIR | MENU " >

   <!ENTITY % attrs -- common attributes for elements --
            "LANG  NAME      #IMPLIED  -- RFC 1766 language tag --
             DIR  (ltr|rtl)  #IMPLIED  -- text directionnality --
             ID      ID      #IMPLIED  -- element identifier
                                          (from RFC1942) --
             CLASS   NAMES   #IMPLIED  -- for subclassing elements
                                          (from RFC1942) --">

   <!ENTITY % just -- an attribute for text justification --
            "ALIGN  (left|right|center|justify)  #IMPLIED"
```

```
          -- default is left for ltr paragraphs, right for rtl -- >

   <!--======= Character mnemonic entities =================-->

   <!ENTITY % ISOlat1 PUBLIC
     "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
   %ISOlat1;

   <!ENTITY amp CDATA "&#38;"     -- ampersand          -->
   <!ENTITY gt CDATA "&#62;"      -- greater than       -->
   <!ENTITY lt CDATA "&#60;"      -- less than          -->
   <!ENTITY quot CDATA "&#34;"    -- double quote       -->

   <!--Entities for language-dependent presentation (BIDI and
       contextual analysis) -->
   <!ENTITY zwnj CDATA "&#8204;"-- zero width non-joiner-->
   <!ENTITY zwj  CDATA "&#8205;"-- zero width joiner-->
   <!ENTITY lrm  CDATA "&#8206;"-- left-to-right mark-->
   <!ENTITY rlm  CDATA "&#8207;"-- right-to-left mark-->


   <!--========= SGML Document Access (SDA) Parameter Entities =====-->

   <!-- HTML contains SGML Document Access (SDA) fixed attributes
   in support of easy transformation to the International Committee
   for Accessible Document Design (ICADD) DTD
         "-//EC-USA-CDA/ICADD//DTD ICADD22//EN".
   ICADD applications are designed to support usable access to
   structured information by print-impaired individuals through
   Braille, large print and voice synthesis.  For more information on
   SDA & ICADD:
           - ISO 12083:1993, Annex A.8, Facilities for Braille,
          large print and computer voice
           - ICADD ListServ
          <ICADD%ASUACAD.BITNET@ARIZVM1.ccit.arizona.edu>
           - Usenet news group bit.listserv.easi
           - Recording for the Blind, +1 800 221 4792
   -->

   <!ENTITY % SDAFORM  "SDAFORM  CDATA  #FIXED"
          -- one to one mapping        -->
   <!ENTITY % SDARULE  "SDARULE  CDATA  #FIXED"
          -- context-sensitive mapping -->
   <!ENTITY % SDAPREF  "SDAPREF  CDATA  #FIXED"
          -- generated text prefix     -->
   <!ENTITY % SDASUFF  "SDASUFF  CDATA  #FIXED"
          -- generated text suffix     -->
   <!ENTITY % SDASUSP  "SDASUSP  NAME   #FIXED"
```

```
          -- suspend transform process -->


   <!--========== Text Markup ====================-->

   <![ %HTML.Highlighting [

   <!ENTITY % font " TT | B | I ">

   <!ENTITY % phrase "EM | STRONG | CODE | SAMP | KBD | VAR | CITE ">

   <!ENTITY % text "#PCDATA|A|IMG|BR|%phrase|%font|SPAN|Q|BDO|SUP|SUB">

   <!ELEMENT (%font;|%phrase) - - (%text)*>
   <!ATTLIST ( TT | CODE | SAMP | KBD | VAR )
           %attrs;
           %SDAFORM; "Lit"
           >

   <!ATTLIST ( B | STRONG )
           %attrs;
           %SDAFORM; "B"
           >
   <!ATTLIST ( I | EM | CITE )
           %attrs;
           %SDAFORM; "It"
           >

   <!-- <TT>       Typewriter text                        -->
   <!-- <B>        Bold text                              -->
   <!-- <I>        Italic text                            -->

   <!-- <EM>       Emphasized phrase                      -->
   <!-- <STRONG>   Strong emphasis                        -->
   <!-- <CODE>     Source code phrase                     -->
   <!-- <SAMP>     Sample text or characters              -->
   <!-- <KBD>      Keyboard phrase, e.g. user input       -->
   <!-- <VAR>      Variable phrase or substitutable       -->
   <!-- <CITE>     Name or title of cited work            -->

   <!ENTITY % pre.content "#PCDATA|A|HR|BR|%font|%phrase|SPAN|BDO">

   ]]>

   <!ENTITY % text "#PCDATA|A|IMG|BR|SPAN|Q|BDO|SUP|SUB">

   <!ELEMENT BR    - O EMPTY>
   <!ATTLIST BR
```

```
          %SDAPREF; "&#RE;"
          >

   <!-- <BR>       Line break      -->

   <!ELEMENT SPAN - - (%text)*>
   <!ATTLIST SPAN
          %attrs;
          %SDAFORM; "other #Attlist"
   >

   <!-- <SPAN>             Generic inline container  -->
   <!-- <SPAN DIR=...>     New counterflow embedding -->
   <!-- <SPAN LANG="..."> Language of contents      -->

   <!ELEMENT Q - - (%text)*>
   <!ATTLIST Q
          %attrs;
          %SDAPREF; '"'
          %SDASUFF; '"'
          >

   <!-- <Q>         Short quotation              -->
   <!-- <Q LANG=xx> Language of quotation is xx  -->
   <!-- <Q DIR=...> New conterflow embedding     -->

   <!ELEMENT BDO - - (%text)+>
   <!ATTLIST BDO
          LANG   NAME      #IMPLIED
          DIR    (ltr|rtl) #REQUIRED
          ID     ID        #IMPLIED
          CLASS  NAMES     #IMPLIED
          %SDAPREF "Bidi Override #Attval(DIR): "
          %SDASUFF "End Bidi"
          >

   <!-- <BDO DIR=...>   Override directionality of text to value of DIR -->
   <!-- <BDO LANG=...>  Language of contents                            -->

   <!ELEMENT (SUP|SUB) - - (#PCDATA)>
   <!ATTLIST (SUP)
          %attrs;
          %SDAPREF "Superscript(#content)"
          >
   <!ATTLIST (SUB)
          %attrs;
          %SDAPREF "Subscript(#content)"
          >
```

```
   <!-- <SUP>      Superscript              -->
   <!-- <SUB>      Subscript                -->

   <!--========= Link Markup ======================-->

   <!ENTITY % linkType "NAMES">

   <!ENTITY % linkExtraAttributes
           "REL %linkType #IMPLIED
           REV %linkType #IMPLIED
           URN CDATA #IMPLIED
           TITLE CDATA #IMPLIED
           METHODS NAMES #IMPLIED
           CHARSET NAME #IMPLIED
           ">

   <![ %HTML.Recommended [
           <!ENTITY % A.content   "(%text)*"

           -- <H1><a name="xxx">Heading</a></H1>
                   is preferred to
              <a name="xxx"><H1>Heading</H1></a>
           -->
   ]]>

   <!ENTITY % A.content   "(%heading|%text)*">

   <!ELEMENT A     - - %A.content -(A)>
   <!ATTLIST A
           %attrs;
           HREF CDATA #IMPLIED
           NAME CDATA #IMPLIED
           %linkExtraAttributes;
           %SDAPREF; "<Anchor: #AttList>"
           >
   <!-- <A>       Anchor; source/destination of link -->
   <!-- <A NAME="..."> Name of this anchor           -->
   <!-- <A HREF="..."> Address of link destination       -->
   <!-- <A URN="...">  Permanent address of destination  -->
   <!-- <A REL=...>    Relationship to destination       -->
   <!-- <A REV=...>    Relationship of destination to this    -->
   <!-- <A TITLE="...">     Title of destination (advisory)        -->
   <!-- <A METHODS="...">   Operations on destination (advisory)   -->
   <!-- <A CHARSET="...">   Charset of destination (advisory)  -->
   <!-- <A LANG="...">     Language of contents btw <A> and </A>   -->
   <!-- <A DIR=...>        Contents is a new counterflow embedding -->

   <!--========== Images ===========================-->
```

```
   <!ELEMENT IMG    - O EMPTY>
   <!ATTLIST IMG
           %attrs;
           SRC CDATA  #REQUIRED
           ALT CDATA #IMPLIED
           ALIGN (top|middle|bottom) #IMPLIED
           ISMAP (ISMAP) #IMPLIED
           %SDAPREF; "<Fig><?SDATrans Img: #AttList>#AttVal(Alt)</Fig>"
           >

   <!-- <IMG>              Image; icon, glyph or illustration      -->
   <!-- <IMG SRC="...">    Address of image object                 -->
   <!-- <IMG ALT="...">    Textual alternative                     -->
   <!-- <IMG ALIGN=...>    Position relative to text               -->
   <!-- <IMG LANG=...>     Image contains "text" in that language  -->
   <!-- <IMG DIR=...>      Inline image acts as a RTL or LTR
                           embedding w/r to BIDI algorithm         -->
   <!-- <IMG ISMAP>        Each pixel can be a link                -->

   <!--========== Paragraphs=======================-->

   <!ELEMENT P     - O (%text)*>
   <!ATTLIST P
           %attrs;
           %just;
           %SDAFORM; "Para"
           >

   <!-- <P>             Paragraph                          -->
   <!-- <P LANG="...">  Language of paragraph text         -->
   <!-- <P DIR=...>     Base directionality of paragraph   -->
   <!-- <P ALIGN=...>   Paragraph alignment (justification) -->

   <!--========== Headings, Titles, Sections ===============-->

   <!ELEMENT HR    - O EMPTY>
   <!ATTLIST HR
           %just;
           %SDAPREF; "&#RE;&#RE;"
           >

   <!-- <HR>       Horizontal rule -->

   <!ELEMENT ( %heading )  - -  (%text;)*>
   <!ATTLIST H1
           %attrs;
           %just;
           %SDAFORM; "H1"
```

```
        >
<!ATTLIST H2
        %attrs;
        %just;
        %SDAFORM; "H2"
        >
<!ATTLIST H3
        %attrs;
        %just;
        %SDAFORM; "H3"
        >
<!ATTLIST H4
        %attrs;
        %just;
        %SDAFORM; "H4"
        >
<!ATTLIST H5
        %attrs;
        %just;
        %SDAFORM; "H5"
        >
<!ATTLIST H6
        %attrs;
        %just;
        %SDAFORM; "H6"
        >

<!-- <H1>       Heading, level 1 -->
<!-- <H2>       Heading, level 2 -->
<!-- <H3>       Heading, level 3 -->
<!-- <H4>       Heading, level 4 -->
<!-- <H5>       Heading, level 5 -->
<!-- <H6>       Heading, level 6 -->


<!--========== Text Flows ======================-->

<![ %HTML.Forms [
        <!ENTITY % block.forms "BLOCKQUOTE | FORM | ISINDEX">
]]>

<!ENTITY % block.forms "BLOCKQUOTE">

<![ %HTML.Deprecated [
        <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>

<!ENTITY % preformatted "PRE">
```

```
   <!ENTITY % block "P | %list | DL
           | %preformatted
           | %block.forms">

   <!ENTITY % flow "(%text|%block)*">

   <!ENTITY % pre.content "#PCDATA | A | HR | BR | SPAN | BDO">
   <!ELEMENT PRE - - (%pre.content)*>
   <!ATTLIST PRE
           %attrs;
           WIDTH NUMBER #implied
           %SDAFORM; "Lit"
           >

   <!-- <PRE>              Preformatted text                    -->
   <!-- <PRE WIDTH=...>    Maximum characters per line          -->
   <!-- <PRE DIR=...>      Base direction of preformatted block -->
   <!-- <PRE LANG=...>     Language of contents                 -->

   <![ %HTML.Deprecated [

   <!ENTITY % literal "CDATA"
           -- historical, non-conforming parsing mode where
              the only markup signal is the end tag
              in full
           -->

   <!ELEMENT (XMP|LISTING) - -  %literal>
   <!ATTLIST XMP
           %attrs;
           %SDAFORM; "Lit"
           %SDAPREF; "Example:&#RE;"
           >
   <!ATTLIST LISTING
           %attrs;
           %SDAFORM; "Lit"
           %SDAPREF; "Listing:&#RE;"
           >

   <!-- <XMP>              Example section         -->
   <!-- <LISTING>          Computer listing        -->

   <!ELEMENT PLAINTEXT - O %literal>
   <!-- <PLAINTEXT>        Plain text passage      -->

   <!ATTLIST PLAINTEXT
           %attrs;
           %SDAFORM; "Lit"
```

```
        >
]]>


<!--========== Lists ==================-->

<!ELEMENT DL    - -  (DT | DD)+>
<!ATTLIST DL
        %attrs;
        COMPACT (COMPACT) #IMPLIED
        %SDAFORM; "List"
        %SDAPREF; "Definition List:"
        >

<!ELEMENT DT    - O (%text)*>
<!ATTLIST DT
        %attrs;
        %SDAFORM; "Term"
        >

<!ELEMENT DD    - O %flow>
<!ATTLIST DD
        %attrs;
        %SDAFORM; "LItem"
        >

<!-- <DL>               Definition list, or glossary    -->
<!-- <DL COMPACT>       Compact style list              -->
<!-- <DT>               Term in definition list         -->
<!-- <DD>               Definition of term              -->

<!ELEMENT (OL|UL) - -  (LI)+>
<!ATTLIST OL
        %attrs;
        %just;
        COMPACT (COMPACT) #IMPLIED
        %SDAFORM; "List"
        >
<!ATTLIST UL
        %attrs;
        %just;
        COMPACT (COMPACT) #IMPLIED
        %SDAFORM; "List"
        >
<!-- <UL>               Unordered list                  -->
<!-- <UL COMPACT>       Compact list style              -->
<!-- <OL>               Ordered, or numbered list       -->
<!-- <OL COMPACT>       Compact list style              -->
```

```
   <!ELEMENT (DIR|MENU) - -  (LI)+ -(%block)>
   <!ATTLIST DIR
           %attrs;
           %just;
           COMPACT (COMPACT) #IMPLIED
           %SDAFORM; "List"
           %SDAPREF; "<LHead>Directory</LHead>"
           >
   <!ATTLIST MENU
           %attrs;
           %just;
           COMPACT (COMPACT) #IMPLIED
           %SDAFORM; "List"
           %SDAPREF; "<LHead>Menu</LHead>"
           >

   <!-- <DIR>              Directory list                -->
   <!-- <DIR COMPACT>      Compact list style            -->
   <!-- <MENU>             Menu list                     -->
   <!-- <MENU COMPACT>     Compact list style            -->

   <!ELEMENT LI    - O %flow>
   <!ATTLIST LI
           %attrs;
           %just;
           %SDAFORM; "LItem"
           >

   <!-- <LI>               List item                     -->

   <!--========== Document Body ====================-->

   <![ %HTML.Recommended [
        <!ENTITY % body.content "(%heading|%block|HR|ADDRESS|IMG)*"
        -- <h1>Heading</h1>
           <p>Text ...
             is preferred to
           <h1>Heading</h1>
           Text ...
        -->
   ]]>

   <!ENTITY % body.content "(%heading | %text | %block |
                        HR | ADDRESS)*">

   <!ELEMENT BODY O O  %body.content>
   <!ATTLIST BODY
           %attrs;
```

```
        >

<!-- <BODY>          Document body                -->
<!-- <BODY DIR=...>  Base direction of whole body -->
<!-- <BODY LANG=...> Language of contents         -->

<!ELEMENT BLOCKQUOTE - - %body.content>
<!ATTLIST BLOCKQUOTE
        %attrs;
        %just;
        %SDAFORM; "BQ"
        >

<!-- <BLOCKQUOTE>       Quoted passage  -->

<!ELEMENT ADDRESS - - (%text|P)*>
<!ATTLIST  ADDRESS
        %attrs;
        %just;
        %SDAFORM; "Lit"
        %SDAPREF; "Address:&#RE;"
        >

<!-- <ADDRESS> Address, signature, or byline -->


<!--======= Forms ====================-->

<![ %HTML.Forms [

<!ELEMENT FORM - - %body.content -(FORM) +(INPUT|SELECT|TEXTAREA)>
<!ATTLIST FORM
        %attrs;
        ACTION CDATA #IMPLIED
        METHOD (%HTTP-Method) GET
        ENCTYPE %Content-Type; "application/x-www-form-urlencoded"
        %SDAPREF; "<Para>Form:</Para>"
        %SDASUFF; "<Para>Form End.</Para>"
        >

<!-- <FORM>                    Fill-out or data-entry form     -->
<!-- <FORM ACTION="...">       Address for completed form      -->
<!-- <FORM METHOD=...>         Method of submitting form       -->
<!-- <FORM ENCTYPE="...">      Representation of form data     -->
<!-- <FORM DIR=...>            Base direction of form          -->
<!-- <FORM LANG=...>           Language of contents            -->

<!ENTITY % InputType "(TEXT | PASSWORD | CHECKBOX |
```

```
                          RADIO | SUBMIT | RESET |
                          IMAGE | HIDDEN | FILE )">
   <!ELEMENT INPUT - O EMPTY>
   <!ATTLIST INPUT
           %attrs;
        TYPE %InputType TEXT
        NAME CDATA #IMPLIED
        VALUE CDATA #IMPLIED
        SRC CDATA #IMPLIED
        CHECKED (CHECKED) #IMPLIED
        SIZE CDATA #IMPLIED
        MAXLENGTH NUMBER #IMPLIED
        ALIGN (top|middle|bottom) #IMPLIED
           ACCEPT CDATA #IMPLIED --list of content types --
           ACCEPT-CHARSET CDATA #IMPLIED --list of charsets accepted --
           %SDAPREF; "Input: "
        >

   <!-- <INPUT>               Form input datum        -->
   <!-- <INPUT TYPE=...>           Type of input interaction    -->
   <!-- <INPUT NAME=...>           Name of form datum           -->
   <!-- <INPUT VALUE="...">   Default/initial/selected value -->
   <!-- <INPUT SRC="...">          Address of image        -->
   <!-- <INPUT CHECKED>            Initial state is "on"        -->
   <!-- <INPUT SIZE=...>           Field size hint         -->
   <!-- <INPUT MAXLENGTH=...>      Data length maximum          -->
   <!-- <INPUT ALIGN=...>          Image alignment         -->
   <!-- <INPUT ACCEPT="...">         List of desired media types    -->
   <!-- <INPUT ACCEPT-CHARSET="..."> List of acceptable charsets    -->

   <!ELEMENT SELECT - - (OPTION+) -(INPUT|SELECT|TEXTAREA)>
   <!ATTLIST SELECT
           %attrs;
           NAME CDATA #REQUIRED
           SIZE NUMBER #IMPLIED
           MULTIPLE (MULTIPLE) #IMPLIED
           %SDAFORM; "List"
           %SDAPREF;
           "<LHead>Select #AttVal(Multiple)</LHead>"
        >

   <!-- <SELECT>            Selection of option(s)        -->
   <!-- <SELECT NAME=...>        Name of form datum       -->
   <!-- <SELECT SIZE=...>        Options displayed at a time   -->
   <!-- <SELECT MULTIPLE>        Multiple selections allowed   -->

   <!ELEMENT OPTION - O (#PCDATA)*>
   <!ATTLIST OPTION
```

```
          %attrs;
          SELECTED (SELECTED) #IMPLIED
          VALUE CDATA #IMPLIED
          %SDAFORM; "LItem"
          %SDAPREF;
          "Option: #AttVal(Value) #AttVal(Selected)"
       >

   <!-- <OPTION>            A selection option       -->
   <!-- <OPTION SELECTED>        Initial state            -->
   <!-- <OPTION VALUE="...">     Form datum value for this option-->

   <!ELEMENT TEXTAREA - - (#PCDATA)* -(INPUT|SELECT|TEXTAREA)>
   <!ATTLIST TEXTAREA
          %attrs;
          NAME CDATA #REQUIRED
          ROWS NUMBER #REQUIRED
          COLS NUMBER #REQUIRED
          ACCEPT-CHARSET CDATA #IMPLIED -- list of charsets accepted --
          %SDAFORM; "Para"
          %SDAPREF; "Input Text -- #AttVal(Name): "
          >

   <!-- <TEXTAREA>               An area for text input        -->
   <!-- <TEXTAREA NAME=...> Name of form datum       -->
   <!-- <TEXTAREA ROWS=...> Height of area           -->
   <!-- <TEXTAREA COLS=...> Width of area            -->

   ]]>


   <!--======= Document Head ======================-->

   <![ %HTML.Recommended [
        <!ENTITY % head.extra "">
   ]]>
   <!ENTITY % head.extra "& NEXTID?">

   <!ENTITY % head.content "TITLE & ISINDEX? & BASE? %head.extra">

   <!ELEMENT HEAD O O  (%head.content) +(META|LINK)>
   <!ATTLIST HEAD
          %attrs;           >

   <!-- <HEAD>     Document head   -->

   <!ELEMENT TITLE - -  (#PCDATA)*  -(META|LINK)>
   <!ATTLIST TITLE
```

```
          %attrs;
          %SDAFORM; "Ti"    >

   <!-- <TITLE>    Title of document -->

   <!ELEMENT LINK - O EMPTY>
   <!ATTLIST LINK
          %attrs;
          HREF CDATA #REQUIRED
          %linkExtraAttributes;
          %SDAPREF; "Linked to : #AttVal (TITLE) (URN) (HREF)>"    >

   <!-- <LINK>         Link from this document            -->
   <!-- <LINK HREF="..."> Address of link destination       -->
   <!-- <LINK URN="..."> Lasting name of destination       -->
   <!-- <LINK REL=...> Relationship to destination       -->
   <!-- <LINK REV=...> Relationship of destination to this    -->
   <!-- <LINK TITLE="..."> Title of destination (advisory)       -->
   <!-- <LINK CHARSET="..."> Charset of destination (advisory)     -->
   <!-- <LINK METHODS="..."> Operations allowed (advisory)         -->

   <!ELEMENT ISINDEX - O EMPTY>
   <!ATTLIST ISINDEX
          %attrs;
          %SDAPREF;
      "<Para>[Document is indexed/searchable.]</Para>">

   <!-- <ISINDEX>          Document is a searchable index         -->

   <!ELEMENT BASE - O EMPTY>
   <!ATTLIST BASE
          HREF CDATA #REQUIRED     >

   <!-- <BASE>             Base context document                -->
   <!-- <BASE HREF="..."> Address for this document             -->

   <!ELEMENT NEXTID - O EMPTY>
   <!ATTLIST NEXTID
          N CDATA #REQUIRED     >

   <!-- <NEXTID>       Next ID to use for link name       -->
   <!-- <NEXTID N=...> Next ID to use for link name       -->

   <!ELEMENT META - O EMPTY>
   <!ATTLIST META
          HTTP-EQUIV  NAME    #IMPLIED
          NAME        NAME    #IMPLIED
          CONTENT     CDATA   #REQUIRED    >
```

```
   <!-- <META>                     Generic Meta-information        -->
   <!-- <META HTTP-EQUIV=...>      HTTP response header name       -->
   <!-- <META NAME=...>          Meta-information name           -->
   <!-- <META CONTENT="...">       Associated information          -->

   <!--======= Document Structure =================-->

   <![ %HTML.Deprecated [
           <!ENTITY % html.content "HEAD, BODY, PLAINTEXT?">
   ]]>
   <!ENTITY % html.content "HEAD, BODY">

   <!ELEMENT HTML O O  (%html.content)>
   <!ENTITY % version.attr "VERSION CDATA #FIXED '%HTML.Version;'">

   <!ATTLIST HTML
           %attrs;
           %version.attr;
           %SDAFORM; "Book"
           >

   <!-- <HTML>              HTML Document  -->
```

7.2. SGML Declaration for HTML

```
   <!SGML  "ISO 8879:1986"
   --
        SGML Declaration for HyperText Markup Language version 2.x
           (HTML 2.x = HTML 2.0 + i18n).

   --

   CHARSET
            BASESET  "ISO Registration Number 177//CHARSET
                      ISO/IEC 10646-1:1993 UCS-4 with
                      implementation level 3//ESC 2/5 2/15 4/6"
            DESCSET  0   9     UNUSED
                     9   2     9
                     11  2     UNUSED
                     13  1     13
                     14  18    UNUSED
                     32  95    32
                     127 1     UNUSED
                     128 32    UNUSED
                     160 2147483486 160
   --
       In ISO 10646, the positions with hexadecimal
       values 0000D800 - 0000DFFF, used in the UTF-16
```

```
       encoding of UCS-4, are reserved, as well as the last
       two code values in each plane of UCS-4, i.e. all
       values of the hexadecimal form xxxxFFFE or xxxxFFFF.
       These code values or the corresponding numeric
       character references must not be included when
       generating a new HTML document, and they should be
       ignored if encountered when processing a HTML
       document.
   --

   CAPACITY        SGMLREF
                   TOTALCAP        150000
                   GRPCAP          150000
              ENTCAP         150000

   SCOPE    DOCUMENT

   SYNTAX
            SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
              17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127

            BASESET  "ISO 646IRV:1991//CHARSET
                      International Reference Version
                      (IRV)//ESC 2/8 4/2"
            DESCSET  0 128 0

            FUNCTION
                     RE            13
                     RS            10
                     SPACE         32
                     TAB SEPCHAR    9

            NAMING   LCNMSTRT ""
                     UCNMSTRT ""
                     LCNMCHAR ".-"
                     UCNMCHAR ".-"
                     NAMECASE GENERAL YES
                              ENTITY  NO
            DELIM    GENERAL  SGMLREF
                     SHORTREF SGMLREF
            NAMES    SGMLREF
            QUANTITY SGMLREF
                     ATTSPLEN 2100
                     LITLEN   1024
                     NAMELEN  72    -- somewhat arbitrary; taken from
                                   internet line length conventions --
                     PILEN    1024
                     TAGLVL   100
```

```
                    TAGLEN   2100
                    GRPGTCNT 150
                    GRPCNT   64

   FEATURES
     MINIMIZE
       DATATAG  NO
       OMITTAG  YES
       RANK     NO
       SHORTTAG YES
     LINK
       SIMPLE   NO
       IMPLICIT NO
       EXPLICIT NO
     OTHER
       CONCUR   NO
       SUBDOC   NO
       FORMAL   YES
     APPINFO    "SDA"  -- conforming SGML Document Access application
                 --
   >
```

7.3. ISO Latin 1 entity set

The following public text lists each of the characters specified in the Added Latin 1 entity set, along with its name, syntax for use, and description. This list is derived from ISO Standard 8879:1986//ENTITIES Added Latin 1//EN. HTML includes the entire entity set, and adds entities for all missing characters in the right part of ISO-8859-1.

```
 <!-- (C) International Organization for Standardization 1986
      Permission to copy in any form is granted for use with
      conforming SGML systems and applications as defined in
      ISO 8879, provided this notice is included in all copies.
   -->
 <!-- Character entity set. Typical invocation:
      <!ENTITY % ISOlat1 PUBLIC
        "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
      %ISOlat1;
   -->
 <!ENTITY nbsp   CDATA " " -- no-break space -->
 <!ENTITY iexcl  CDATA "&#161;" -- inverted exclamation mark -->
 <!ENTITY cent   CDATA "&#162;" -- cent sign -->
 <!ENTITY pound  CDATA "&#163;" -- pound sterling sign -->
 <!ENTITY curren CDATA "&#164;" -- general currency sign -->
 <!ENTITY yen    CDATA "&#165;" -- yen sign -->
 <!ENTITY brvbar CDATA "&#166;" -- broken (vertical) bar -->
```

```
    <!ENTITY sect   CDATA "&#167;" -- section sign -->
    <!ENTITY uml    CDATA "&#168;" -- umlaut (dieresis) -->
    <!ENTITY copy   CDATA "&#169;" -- copyright sign -->
    <!ENTITY ordf   CDATA "&#170;" -- ordinal indicator, feminine -->
    <!ENTITY laquo  CDATA "&#171;" -- angle quotation mark, left -->
    <!ENTITY not    CDATA "&#172;" -- not sign -->
    <!ENTITY shy    CDATA "&#173;" -- soft hyphen -->
    <!ENTITY reg    CDATA "&#174;" -- registered sign -->
    <!ENTITY macr   CDATA "&#175;" -- macron -->
    <!ENTITY deg    CDATA "&#176;" -- degree sign -->
    <!ENTITY plusmn CDATA "&#177;" -- plus-or-minus sign -->
    <!ENTITY sup2   CDATA "&#178;" -- superscript two -->
    <!ENTITY sup3   CDATA "&#179;" -- superscript three -->
    <!ENTITY acute  CDATA "&#180;" -- acute accent -->
    <!ENTITY micro  CDATA "&#181;" -- micro sign -->
    <!ENTITY para   CDATA "&#182;" -- pilcrow (paragraph sign) -->
    <!ENTITY middot CDATA "&#183;" -- middle dot -->
    <!ENTITY cedil  CDATA "&#184;" -- cedilla -->
    <!ENTITY sup1   CDATA "&#185;" -- superscript one -->
    <!ENTITY ordm   CDATA "&#186;" -- ordinal indicator, masculine -->
    <!ENTITY raquo  CDATA "&#187;" -- angle quotation mark, right -->
    <!ENTITY frac14 CDATA "&#188;" -- fraction one-quarter -->
    <!ENTITY frac12 CDATA "&#189;" -- fraction one-half -->
    <!ENTITY frac34 CDATA "&#190;" -- fraction three-quarters -->
    <!ENTITY iquest CDATA "&#191;" -- inverted question mark -->
    <!ENTITY Agrave CDATA "&#192;" -- capital A, grave accent -->
    <!ENTITY Aacute CDATA "&#193;" -- capital A, acute accent -->
    <!ENTITY Acirc  CDATA "&#194;" -- capital A, circumflex accent -->
    <!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
    <!ENTITY Auml   CDATA "&#196;" -- capital A, dieresis or umlaut -->
    <!ENTITY Aring  CDATA "&#197;" -- capital A, ring -->
    <!ENTITY AElig  CDATA "&#198;" -- capital AE diphthong (ligature) -->
    <!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
    <!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
    <!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
    <!ENTITY Ecirc  CDATA "&#202;" -- capital E, circumflex accent -->
    <!ENTITY Euml   CDATA "&#203;" -- capital E, dieresis or umlaut -->
    <!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
    <!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
    <!ENTITY Icirc  CDATA "&#206;" -- capital I, circumflex accent -->
    <!ENTITY Iuml   CDATA "&#207;" -- capital I, dieresis or umlaut -->
    <!ENTITY ETH    CDATA "&#208;" -- capital Eth, Icelandic -->
    <!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
    <!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
    <!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
    <!ENTITY Ocirc  CDATA "&#212;" -- capital O, circumflex accent -->
    <!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
    <!ENTITY Ouml   CDATA "&#214;" -- capital O, dieresis or umlaut -->
```

```
    <!ENTITY times  CDATA "&#215;" -- multiply sign -->
    <!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
    <!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
    <!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
    <!ENTITY Ucirc  CDATA "&#219;" -- capital U, circumflex accent -->
    <!ENTITY Uuml   CDATA "&#220;" -- capital U, dieresis or umlaut -->
    <!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
    <!ENTITY THORN  CDATA "&#222;" -- capital Thorn, Icelandic -->
    <!ENTITY szlig  CDATA "&#223;" -- small sharp s, German (sz ligature) -->
    <!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
    <!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
    <!ENTITY acirc  CDATA "&#226;" -- small a, circumflex accent -->
    <!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
    <!ENTITY auml   CDATA "&#228;" -- small a, dieresis or umlaut -->
    <!ENTITY aring  CDATA "&#229;" -- small a, ring -->
    <!ENTITY aelig  CDATA "&#230;" -- small ae diphthong (ligature) -->
    <!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
    <!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
    <!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
    <!ENTITY ecirc  CDATA "&#234;" -- small e, circumflex accent -->
    <!ENTITY euml   CDATA "&#235;" -- small e, dieresis or umlaut -->
    <!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
    <!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
    <!ENTITY icirc  CDATA "&#238;" -- small i, circumflex accent -->
    <!ENTITY iuml   CDATA "&#239;" -- small i, dieresis or umlaut -->
    <!ENTITY eth    CDATA "&#240;" -- small eth, Icelandic -->
    <!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
    <!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
    <!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
    <!ENTITY ocirc  CDATA "&#244;" -- small o, circumflex accent -->
    <!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
    <!ENTITY ouml   CDATA "&#246;" -- small o, dieresis or umlaut -->
    <!ENTITY divide CDATA "&#247;" -- divide sign -->
    <!ENTITY oslash CDATA "&#248;" -- small o, slash -->
    <!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
    <!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
    <!ENTITY ucirc  CDATA "&#251;" -- small u, circumflex accent -->
    <!ENTITY uuml   CDATA "&#252;" -- small u, dieresis or umlaut -->
    <!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
    <!ENTITY thorn  CDATA "&#254;" -- small thorn, Icelandic -->
    <!ENTITY yuml   CDATA "&#255;" -- small y, dieresis or umlaut -->
```

8. Security Considerations

Anchors, embedded images, and all other elements which contain URIs as parameters may cause the URI to be dereferenced in response to user input. In this case, the security considerations of [RFC1738] apply.

The widely deployed methods for submitting form requests -- HTTP and SMTP -- provide little assurance of confidentiality. Information providers who request sensitive information via forms -- especially by way of the 'PASSWORD' type input field (see section 8.1.2 in [RFC1866]) -- should be aware and make their users aware of the lack of confidentiality.

Bibliography

[BRYAN88] M. Bryan, "SGML -- An Author's Guide to the Standard Generalized Markup Language", Addison-Wesley, Reading, 1988.

[ERCS] Extended Reference Concrete Syntax for SGML. <http://www.sgmlopen.org/sgml/docs/ercs/ercs-home.html>

[GOLD90] C. F. Goldfarb, "The SGML Handbook", Y. Rubinsky, Ed., Oxford University Press, 1990.

[HTTP-1.1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

[ISO-639] ISO 639:1988. International standard -- Code for the representation of the names of languages. Technical content in <http://www.sil.org/sgml/iso639a.html>

[ISO-8859] ISO 8859. International standard -- Information processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1 (1987) -- Part 2: Latin alphabet No. 2 (1987) -- Part 3: Latin alphabet No. 3 (1988) -- Part 4: Latin alphabet No. 4 (1988) -- Part 5: Latin/Cyrillic alphabet (1988) -- Part 6: Latin/Arabic alphabet (1987) -- Part : Latin/Greek alphabet (1987) -- Part 8: Latin/Hebrew alphabet (1988) -- Part 9: Latin alphabet No. 5 (1989) -- Part 10: Latin alphabet No. 6 (1992)

[ISO-8879] ISO 8879:1986. International standard -- Information processing -- Text and office systems -- Standard generalized markup language (SGML).

[ISO-10646] ISO/IEC 10646-1:1993. International standard -- Information technology -- Universal multiple-octet coded character Sset (UCS) -- Part 1: Architecture and basic multilingual plane.

[NICOL] G.T. Nicol, "The Multilingual World Wide Web", Electronic Book Technologies, 1995, <http://www.ebt.com/docs/multling.html>

[NICOL2] G.T. Nicol, "MIME Header Supplemented File Type", Work in Progress, EBT, October 1995.

[RFC1345] Simonsen, K., "Character Mnemonics & Character Sets", RFC 1345, Rationel Almen Planlaegning, June 1992.

[RFC1468] Murai, J., Crispin M., and E. van der Poel, "Japanese Character Encoding for Internet Messages", RFC 1468, Keio University, Panda Programming, June 1993.

[RFC2045] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, Innosoft, First Virtual, November 1996.

[RFC1641] Goldsmith, D., and M.Davis, "Using Unicode with MIME", RFC 1641, Taligent inc., July 1994.

[RFC1642] Goldsmith, D., and M. Davis, "UTF-7: A Mail-safe Transformation Format of Unicode", RFC 1642, Taligent, Inc., July 1994.

[RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox PARC, University of Minnesota, October 1994.

[RFC1766] Alverstrand, H., "Tags for the Identification of Languages", RFC 1766, UNINETT, March 1995.

[RFC1866] Berners-Lee, T., and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, MIT/W3C, November 1995.

[RFC1867] Nebel, E., and L. Masinter, "Form-based File Upload in HTML", RFC 1867, Xerox Corporation, November 1995.

[RFC1942] Raggett, D., "HTML Tables", RFC 1942, W3C, May 1996.

[RFC2068] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

[SQ91] SoftQuad, "The SGML Primer", 3rd ed., SoftQuad Inc., 1991.

[TAKADA] Toshihiro Takada, "Multilingual Information Exchange through the World-Wide Web", Computer Networks and ISDN Systems, Vol. 27, No. 2, Nov. 1994 , p. 235-241.

[TEI] TEI Guidelines for Electronic Text Encoding and Interchange. <http://etext.virgina.edu/TEI.html>

[UNICODE] The Unicode Consortium, "The Unicode Standard -- Worldwide Character Encoding -- Version 1.0", Addison-Wesley, Volume 1, 1991, Volume 2, 1992, and Technical Report #4, 1993. The BIDI algorithm is in appendix A of volume 1, with corrections in appendix D of volume 2.

[UTF-8] ISO/IEC 10646-1:1993 AMENDMENT 2 (1996). UCS Transformation Format 8 (UTF-8).

[VANH90] E. van Hervijnen, "Practical SGML", Kluwer Academicq Publishers Group, Norwell and Dordrecht, 1990.

Authors' Addresses

Frangois Yergeau
Alis Technologies
100, boul. Alexis-Nihon, bureau 600
Montrial QC H4M 2P2
Canada

Tel: +1 (514) 747-2547
Fax: +1 (514) 747-2561
EMail: fyergeau@alis.com

Gavin Thomas Nicol
Electronic Book Technologies, Japan
1-29-9 Tsurumaki,
Setagaya-ku,
Tokyo
Japan

Tel: +81-3-3230-8161
Fax: +81-3-3230-8163
EMail: gtn@ebt.com, gtn@twics.co.jp

Glenn Adams
Spyglass
118 Magazine Street
Cambridge, MA 02139
U.S.A.

Tel: +1 (617) 864-5524
Fax: +1 (617) 864-4965
EMail: glenn@spyglass.com

Martin J. Duerst
Multimedia-Laboratory
Department of Computer Science
University of Zurich
Winterthurerstrasse 190
CH-8057 Zurich
Switzerland

Tel: +41 1 257 43 16
Fax: +41 1 363 00 35
EMail: mduerst@ifi.unizh.ch