                            XML Media Types

Status of this Memo

Copyright Notice

Abstract

   This document proposes two new media subtypes, text/xml and
   application/xml, for use in exchanging network entities which are
   conforming Extensible Markup Language (XML). XML entities are
   currently exchanged via the HyperText Transfer Protocol on the World
   Wide Web, are an integral part of the WebDAV protocol for remote web
   authoring, and are expected to have utility in many domains.

Table of Contents

1 Introduction

   The World Wide Web Consortium (W3C) has issued a Recommendation
   [REC-XML] which defines the Extensible Markup Language (XML), version
   1. To enable the exchange of XML network entities, this document
   proposes two new media types, text/xml and application/xml.

   XML entities are currently exchanged on the World Wide Web, and XML
   is also used for property values and parameter marshalling by the
   WebDAV protocol for remote web authoring. Thus, there is a need for a
   media type to properly label the exchange of XML network entities.
   (Note that, as sometimes happens between two communities, both MIME
   and XML have defined the term entity, with different meanings.)

   Although XML is a subset of the Standard Generalized Markup Language
   (SGML) [ISO-8897], and currently is assigned the media types
   text/sgml and application/sgml, there are several reasons why use of
   text/sgml or application/sgml to label XML is inappropriate. First,
   there exist many applications which can process XML, but which cannot
   process SGML, due to SGML's larger feature set. Second, SGML
   applications cannot always process XML entities, because XML uses
   features of recent technical corrigenda to SGML. Third, the
   definition of text/sgml and application/sgml [RFC-1874] includes
   parameters for SGML bit combination transformation format (SGML-
   bctf), and SGML boot attribute (SGML-boot). Since XML does not use
   these parameters, it would be ambiguous if such parameters were given
   for an XML entity. For these reasons, the best approach for labeling
   XML network entities is to provide new media types for XML.

   Since XML is an integral part of the WebDAV Distributed Authoring
   Protocol, and since World Wide Web Consortium Recommendations have
   conventionally been assigned IETF tree media types, and since similar
   media types (HTML, SGML) have been assigned IETF tree media types,
   the XML media types also belong in the IETF media types tree.

2 Notational Conventions

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC-2119].

3 XML Media Types

   This document introduces two new media types for XML entities,
   text/xml and application/xml. Registration information for these
   media types are described in the sections below.

   Every XML entity is suitable for use with the application/xml media
   type without modification. But this does not exploit the fact that
   XML can be treated as plain text in many cases. MIME user agents
   (and web user agents) that do not have explicit support for
   application/xml will treat it as application/octet-stream, for
   example, by offering to save it to a file.

   To indicate that an XML entity should be treated as plain text by
   default, use the text/xml media type. This restricts the encoding
   used in the XML entity to those that are compatible with the
   requirements for text media types as described in [RFC-2045] and
   [RFC-2046], e.g., UTF-8, but not UTF-16 (except for HTTP).

   XML provides a general framework for defining sequences of structured
   data. In some cases, it may be desirable to define new media types
   which use XML but define a specific application of XML, perhaps due
   to domain-specific security considerations or runtime information.
   This document does not prohibit future media types dedicated to such
   XML applications. However, developers of such media types are
   recommended to use this document as a basis. In particular, the
   charset parameter should be used in the same manner.

   Within the XML specification, XML entities can be classified into
   four types. In the XML terminology, they are called "document
   entities", "external DTD subsets", "external parsed entities", and
   "external parameter entities". The media types text/xml and
   application/xml can be used for any of these four types.

3.1 Text/xml Registration

   MIME media type name: text

   MIME subtype name: xml

   Mandatory parameters: none

   Optional parameters: charset

      Although listed as an optional parameter, the use of the charset
      parameter is STRONGLY RECOMMENDED, since this information can be
      used by XML processors to determine authoritatively the character
      encoding of the XML entity. The charset parameter can also be used
      to provide protocol-specific operations, such as charset-based
      content negotiation in HTTP. "UTF-8" [RFC-2279] is the
      recommended value, representing the UTF-8 charset. UTF-8 is
      supported by all conforming XML processors [REC-XML].

      If the XML entity is transmitted via HTTP, which uses a MIME-like
      mechanism that is exempt from the restrictions on the text top-
      level type (see section 19.4.1 of HTTP 1.1 [RFC-2068]), "UTF-16"
      (Appendix C.3 of [UNICODE] and Amendment 1 of [ISO-10646]) is also
      recommended. UTF-16 is supported by all conforming XML processors
      [REC-XML]. Since the handling of CR, LF and NUL for text types in
      most MIME applications would cause undesired transformations of
      individual octets in UTF-16 multi-octet characters, gateways from
      HTTP to these MIME applications MUST transform the XML entity from
      a text/xml; charset="utf-16" to application/xml; charset="utf-16".

      Conformant with [RFC-2046], if a text/xml entity is received with
      the charset parameter omitted, MIME processors and XML processors
      MUST use the default charset value of "us-ascii". In cases where
      the XML entity is transmitted via HTTP, the default charset value
      is still "us-ascii".

      Since the charset parameter is authoritative, the charset is not
      always declared within an XML encoding declaration. Thus, special
      care is needed when the recipient strips the MIME header and
      provides persistent storage of the received XML entity (e.g., in a
      file system). Unless the charset is UTF-8 or UTF-16, the recipient
      SHOULD also persistently store information about the charset,
      perhaps by embedding a correct XML encoding declaration within the
      XML entity.

   Encoding considerations:

      This media type MAY be encoded as appropriate for the charset and
      the capabilities of the underlying MIME transport. For 7-bit
      transports, data in both UTF-8 and UTF-16 is encoded in quoted-
      printable or base64. For 8-bit clean transport (e.g., ESMTP,
      8BITMIME, or NNTP), UTF-8 is not encoded, but UTF-16 is base64
      encoded. For binary clean transports (e.g., HTTP), no content-
      transfer-encoding is necessary.

Security considerations:

    See section 4 below.

Interoperability considerations:

    XML has proven to be interoperable across WebDAV clients and
    servers, and for import and export from multiple XML authoring
    tools.

Published specification: see [REC-XML]

Applications which use this media type:

    XML is device-, platform-, and vendor-neutral and is supported by
    a wide range of Web user agents, WebDAV clients and servers, as
    well as XML authoring tools.

Additional information:

    Magic number(s): none

    Although no byte sequences can be counted on to always be present,
    XML entities in ASCII-compatible charsets (including UTF-8) often
    begin with hexadecimal 3C 3F 78 6D 6C ("<?xml"). For more
    information, see Appendix F of [REC-XML].

    File extension(s): .xml, .dtd
    Macintosh File Type Code(s): "TEXT"

Person & email address for further information:

    Dan Connolly <connolly@w3.org>
    Murata Makoto (Family Given) <murata@fxis.fujixerox.co.jp>

Intended usage: COMMON

Author/Change controller:

    The XML specification is a work product of the World Wide Web
    Consortium's XML Working Group, and was edited by:

    Tim Bray <tbray@textuality.com>
    Jean Paoli <jeanpa@microsoft.com>
    C. M. Sperberg-McQueen <cmsmcq@uic.edu>

    The W3C, and the W3C XML working group, has change control over
    the XML specification.

3.2 Application/xml Registration

   MIME media type name: application

   MIME subtype name: xml

   Mandatory parameters: none

   Optional parameters: charset

      Although listed as an optional parameter, the use of the charset
      parameter is STRONGLY RECOMMENDED, since this information can be
      used by XML processors to determine authoritatively the charset of
      the XML entity. The charset parameter can also be used to provide
      protocol-specific operations, such as charset-based content
      negotiation in HTTP.

      "UTF-8" [RFC-2279] and "UTF-16" (Appendix C.3 of [UNICODE] and
      Amendment 1 of [ISO-10646]) are the recommended values,
      representing the UTF-8 and UTF-16 charsets, respectively. These
      charsets are preferred since they are supported by all conforming
      XML processors [REC-XML].

      If an application/xml entity is received where the charset
      parameter is omitted, no information is being provided about the
      charset by the MIME Content-Type header. Conforming XML processors
      MUST follow the requirements in section 4.3.3 of [REC-XML] which
      directly address this contingency. However, MIME processors which
      are not XML processors should not assume a default charset if the
      charset parameter is omitted from an application/xml entity.

      Since the charset parameter is authoritative, the charset is not
      always declared within an XML encoding declaration. Thus, special
      care is needed when the recipient strips the MIME header and
      provides persistent storage of the received XML entity (e.g., in a
      file system). Unless the charset is UTF-8 or UTF-16, the
      recipient SHOULD also persistently store information about the
      charset, perhaps by embedding a correct XML encoding declaration
      within the XML entity.

Encoding considerations:

   This media type MAY be encoded as appropriate for the charset and
   the capabilities of the underlying MIME transport. For 7-bit
   transports, data in both UTF-8 and UTF-16 is encoded in quoted-
   printable or base64. For 8-bit clean transport (e.g., ESMTP,
   8BITMIME, or NNTP), UTF-8 is not encoded, but UTF-16 is base64
   encoded. For binary clean transport (e.g., HTTP), no content-
   transfer-encoding is necessary.

Security considerations:

   See section 4 below.

Interoperability considerations:

   XML has proven to be interoperable for import and export from
   multiple XML authoring tools.

Published specification: see [REC-XML]

Applications which use this media type:

   XML is device-, platform-, and vendor-neutral and is supported by
   a wide range of Web user agents and XML authoring tools.

Additional information:

   Magic number(s): none

   Although no byte sequences can be counted on to always be present,
   XML entities in ASCII-compatible charsets (including UTF-8) often
   begin with hexadecimal 3C 3F 78 6D 6C ("<?xml"), and those in
   UTF-16 often begin with hexadecimal FE FF 00 3C 00 3F 00 78 00 6D
   or FF FE 3C 00 3F 00 78 00 6D 00 (the Byte Order Mark (BOM)
   followed by "<?xml"). For more information, see Annex F of [REC-
   XML].

   File extension(s): .xml, .dtd
   Macintosh File Type Code(s): "TEXT"

Person & email address for further information:

   Dan Connolly <connolly@w3.org>
   Murata Makoto (Family Given) <murata@fxis.fujixerox.co.jp>

Intended usage: COMMON

Author/Change controller:

    The XML specification is a work product of the World Wide Web
    Consortium's XML Working Group, and was edited by:

    Tim Bray <tbray@textuality.com>
    Jean Paoli <jeanpa@microsoft.com>
    C. M. Sperberg-McQueen <cmsmcq@uic.edu>

    The W3C, and the W3C XML working group, has change control over
    the XML specification.

4 Security Considerations

    XML, as a subset of SGML, has the same security considerations as
    specified in [RFC-1874].

    To paraphrase section 3 of [RFC-1874], XML entities contain
    information to be parsed and processed by the recipient's XML system.
    These entities may contain and such systems may permit explicit
    system level commands to be executed while processing the data. To
    the extent that an XML system will execute arbitrary command strings,
    recipients of XML entities may be at risk. In general, it may be
    possible to specify commands that perform unauthorized file
    operations or make changes to the display processor's environment
    that affect subsequent operations.

    Use of XML is expected to be varied, and widespread. XML is under
    scrutiny by a wide range of communities for use as a common syntax
    for community-specific metadata. For example, the Dublin Core group
    is using XML for document metadata, and a new effort has begun which
    is considering use of XML for medical information. Other groups view
    XML as a mechanism for marshalling parameters for remote procedure
    calls. More uses of XML will undoubtedly arise.

    Security considerations will vary by domain of use. For example, XML
    medical records will have much more stringent privacy and security
    considerations than XML library metadata. Similarly, use of XML as a
    parameter marshalling syntax necessitates a case by case security
    review.

    XML may also have some of the same security concerns as plain text.
    Like plain text, XML can contain escape sequences which, when
    displayed, have the potential to change the display processor
    environment in ways that adversely affect subsequent operations.
    Possible effects include, but are not limited to, locking the
    keyboard, changing display parameters so subsequent displayed text is
    unreadable, or even changing display parameters to deliberately

obscure or distort subsequent displayed material so that its meaning
is lost or altered. Display processors should either filter such
material from displayed text or else make sure to reset all important
settings after a given display operation is complete.

Some terminal devices have keys whose output, when pressed, can be
changed by sending the display processor a character sequence. If
this is possible the display of a text object containing such
character sequences could reprogram keys to perform some illicit or
dangerous action when the key is subsequently pressed by the user.
In some cases not only can keys be programmed, they can be triggered
remotely, making it possible for a text display operation to directly
perform some unwanted action. As such, the ability to program keys
should be blocked either by filtering or by disabling the ability to
program keys entirely.

Note that it is also possible to construct XML documents which make
use of what XML terms "entity references" (using the XML meaning of
the term "entity", which differs from the MIME definition of this
term), to construct repeated expansions of text. Recursive expansions
are prohibited [REC-XML] and XML processors are required to detect
them. However, even non-recursive expansions may cause problems with
the finite computing resources of computers, if they are performed
many times.

5   The Byte Order Mark (BOM) and Conversions to/from UTF-16

The XML Recommendation, in section 4.3.3, specifies that UTF-16 XML
entities must begin with a byte order mark (BOM), which is the ZERO
WIDTH NO-BREAK SPACE character, hexadecimal sequence 0xFEFF (or
0xFFFE, depending on endian). The XML Recommendation further states
that the BOM is an encoding signature, and is not part of either the
markup or the character data of the XML document.

Due to the BOM, applications which convert XML from the UTF-16
encoding to another encoding SHOULD strip the BOM before conversion.
Similarly, when converting from another encoding into UTF-16, the BOM
SHOULD be added after conversion is complete.

6   Examples

The examples below give the value of the Content-type MIME header and
the XML declaration (which includes the encoding declaration) inside
the XML entity. For UTF-16 examples, the Byte Order Mark character
is denoted as "{BOM}", and the XML declaration is assumed to come at
the beginning of the XML entity, immediately following the BOM. Note
that other MIME headers may be present, and the XML entity may

contain other data in addition to the XML declaration; the examples
focus on the Content-type header and the encoding declaration for
clarity.

6.1 text/xml with UTF-8 Charset

   Content-type: text/xml; charset="utf-8"

   <?xml version="1.0" encoding="utf-8"?>

   This is the recommended charset value for use with text/xml. Since
   the charset parameter is provided, MIME and XML processors must treat
   the enclosed entity as UTF-8 encoded.

   If sent using a 7-bit transport (e.g. SMTP), the XML entity must use
   a content-transfer-encoding of either quoted-printable or base64.
   For an 8-bit clean transport (e.g., ESMTP, 8BITMIME, or NNTP), or a
   binary clean transport (e.g., HTTP) no content-transfer-encoding is
   necessary.

6.2 text/xml with UTF-16 Charset

   Content-type: text/xml; charset="utf-16"

   {BOM}<?xml version='1.0' encoding='utf-16'?>

   This is possible only when the XML entity is transmitted via HTTP,
   which uses a MIME-like mechanism and is a binary-clean protocol,
   hence does not perform CR and LF transformations and allows NUL
   octets. This differs from typical text MIME type processing (see
   section 19.4.1 of HTTP 1.1 [RFC-2068] for details).

   Since HTTP is binary clean, no content-transfer-encoding is
   necessary.

6.3 text/xml with ISO-2022-KR Charset

   Content-type: text/xml; charset="iso-2022-kr"

   <?xml version="1.0" encoding='iso-2022-kr'?>

   This example shows text/xml with a Korean charset (e.g., Hangul)
   encoded following the specification in [RFC-1557]. Since the charset
   parameter is provided, MIME and XML processors must treat the
   enclosed entity as encoded per [RFC-1557].

   Since ISO-2022-KR has been defined to use only 7 bits of data, no
   content-transfer-encoding is necessary with any transport.

6.4 text/xml with Omitted Charset

   Content-type: text/xml

   {BOM}<?xml version="1.0" encoding="utf-16"?>

   This example shows text/xml with the charset parameter omitted. In
   this case, MIME and XML processors must assume the charset is "us-
   ascii", the default charset value for text media types specified in
   [RFC-2046]. The default of "us-ascii" holds even if the text/xml
   entity is transported using HTTP.

   Omitting the charset parameter is NOT RECOMMENDED for text/xml. For
   example, even if the contents of the XML entity are UTF-16 or UTF-8,
   or the XML entity has an explicit encoding declaration, XML and MIME
   processors must assume the charset is "us-ascii".

6.5 application/xml with UTF-16 Charset

   Content-type: application/xml; charset="utf-16"

   {BOM}<?xml version="1.0"?>

   This is a recommended charset value for use with application/xml.
   Since the charset parameter is provided, MIME and XML processors must
   treat the enclosed entity as UTF-16 encoded.

   If sent using a 7-bit transport (e.g., SMTP) or an 8-bit clean
   transport (e.g., ESMTP, 8BITMIME, or NNTP), the XML entity must be
   encoded in quoted-printable or base64. For a binary clean transport
   (e.g., HTTP), no content-transfer-encoding is necessary.

6.6 application/xml with ISO-2022-KR Charset

   Content-type: application/xml; charset="iso-2022-kr"

   <?xml version="1.0" encoding="iso-2022-kr"?>

   This example shows application/xml with a Korean charset (e.g.,
   Hangul) encoded following the specification in [RFC-1557]. Since the
   charset parameter is provided, MIME and XML processors must treat the
   enclosed entity as encoded per [RFC-1557], independent of whether the
   XML entity has an internal encoding declaration (this example does
   show such a declaration, which agrees with the charset parameter).

   Since ISO-2022-KR has been defined to use only 7 bits of data, no
   content-transfer-encoding is necessary with any transport.

6.7 application/xml with Omitted Charset and UTF-16 XML Entity

    Content-type: application/xml

    {BOM}<?xml version='1.0'?>

    For this example, the XML entity begins with a BOM. Since the
    charset has been omitted, a conforming XML processor follows the
    requirements of [REC-XML], section 4.3.3. Specifically, the XML
    processor reads the BOM, and thus knows deterministically that the
    charset encoding is UTF-16.

    An XML-unaware MIME processor should make no assumptions about the
    charset of the XML entity.

6.8 application/xml with Omitted Charset and UTF-8 Entity

    Content-type: application/xml

    <?xml version='1.0'?>

    In this example, the charset parameter has been omitted, and there is
    no BOM. Since there is no BOM, the XML processor follows the
    requirements in section 4.3.3, and optionally applies the mechanism
    described in appendix F (which is non-normative) of [REC-XML] to
    determine the charset encoding of UTF-8. The XML entity does not
    contain an encoding declaration, but since the encoding is UTF-8,
    this is still a conforming XML entity.

    An XML-unaware MIME processor should make no assumptions about the
    charset of the XML entity.

6.9 application/xml with Omitted Charset and Internal Encoding
     Declaration

    Content-type: application/xml

    <?xml version='1.0' encoding="ISO-10646-UCS-4"?>

    In this example, the charset parameter has been omitted, and there is
    no BOM. However, the XML entity does have an encoding declaration
    inside the XML entity which specifies the entity's charset. Following
    the requirements in section 4.3.3, and optionally applying the
    mechanism described in appendix F (non-normative) of [REC-XML], the
    XML processor determines the charset encoding of the XML entity (in
    this example, UCS-4).

An XML-unaware MIME processor should make no assumptions about the
charset of the XML entity.

7 References

   [ISO-10646] ISO/IEC, Information Technology - Universal Multiple-
                Octet Coded Character Set (UCS) - Part 1: Architecture
                and Basic Multilingual Plane, May 1993.

   [ISO-8897]   ISO (International Organization for Standardization) ISO
                8879:1986(E) Information Processing -- Text and Office
                Systems -- Standard Generalized Markup Language (SGML).
                First edition -- 1986- 10-15.

   [REC-XML]   T. Bray, J. Paoli, C. M. Sperberg-McQueen, "Extensible
                Markup Language (XML)" World Wide Web Consortium
                Recommendation REC- xml-19980210.
                http://www.w3.org/TR/1998/REC-xml-19980210.

   [RFC-1557]   Choi, U., Chon, K., and H. Park. "Korean Character
                Encoding for Internet Messages", RFC 1557. December,
                1993.

   [RFC-1874]   Levinson, E., "SGML Media Types", RFC 1874. December
                1995.

   [RFC-2119]   Bradner, S., "Key words for use in RFCs to Indicate
                Requirement Levels", BCP 14, RFC 2119, March 1997.

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

   [RFC-2046]   Freed, N., and N. Borenstein, "Multipurpose Internet Mail
                Extensions (MIME) Part Two: Media Types", RFC 2046,
                November 1996.

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

   [RFC-2279]   Yergeau, F., "UTF-8, a transformation format of ISO
                10646", RFC 2279, January 1998.

   [UNICODE]   The Unicode Consortium, "The Unicode Standard -- Version
                2.0", Addison-Wesley, 1996.

8 Acknowledgements

9 Addresses of Authors

   E. James Whitehead, Jr.
   Dept. of Information and Computer Science
   University of California, Irvine
   Irvine, CA 92697-3425

   EMail: ejw@ics.uci.edu

   Murata Makoto (Family Given)
   Fuji Xerox Information Systems,
   KSP 9A7, 2-1, Sakado 3-chome, Takatsu-ku,
   Kawasaki-shi, Kanagawa-ken,
   213 Japan

   EMail: murata@fxis.fujixerox.co.jp

10 Full Copyright Statement