Internet Engineering Task Force (IETF)
Request for Comments: 6839
Updates: 3023
Category: Informational
ISSN: 2070-1721
January 2013

T. Hansen
AT&T Laboratories
A. Melnikov
Isode Ltd

Additional Media Type Structured Syntax Suffixes

Abstract

A content media type name sometimes includes partitioned meta-information distinguished by a structured syntax to permit noting an attribute of the media as a suffix to the name. This document defines several structured syntax suffixes for use with media type registrations. In particular, it defines and registers the "+json", "+ber", "+der", "+fastinfoset", "+wbxml" and "+zip" structured syntax suffixes, and provides a media type structured syntax suffix registration form for the "+xml" structured syntax suffix.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6839.

Copyright Notice

Table of Contents

1. Introduction

[RFC3023] created the +xml suffix convention that can be used when defining names for media types whose representation uses XML underneath. That is, they could have been successfully parsed as if the media type had been application/xml in addition to their being parsed as their media type that is using the +xml suffix. [RFC6838] defines the media type "Structured Syntax Suffix Registry" to be used for such structured syntax suffixes.

A variety of structured syntax suffixes have already been used in some media type registrations, in particular "+json", "+der", "+fastinfoset", and "+wbxml". This document defines and registers these structured syntax suffixes in the Structured Syntax Suffix Registry, along with "+ber" and "+zip". In addition, this document updates [RFC3023] to formally register the "+xml" structured syntax suffix according to the procedure defined in [RFC6838].

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 [RFC2119].

2. When to Use These Structured Syntax Suffixes

Each of the structured syntax suffixes defined in this document is appropriate for use when the media type identifies the semantics of the protocol payload. That is, knowing the semantics of the specific media type provides for more specific processing of the content than that afforded by generic processing of the underlying representation.

At the same time, using the suffix allows receivers of the media types to do generic processing of the underlying representation in cases where

> they do not need to perform special handling of the particular semantics of the exact media type, and
>
> there is no special knowledge needed by such a generic processor in order to parse that underlying representation other than what would be needed to parse any example of that underlying representation.

3. Initial Structured Syntax Suffix Definitions

3.1. The +json Structured Syntax Suffix

[RFC4627] defines the "application/json" media type. The suffix "+json" MAY be used with any media type whose representation follows that established for "application/json". The media type structured syntax suffix registration form follows. See [RFC6838] for definitions of each of the registration form headings.

Name: JavaScript Object Notation (JSON)

+suffix: +json

References: [RFC4627]

Encoding considerations:

> Per [RFC4627], JSON is allowed to be represented using UTF-8, UTF-16, or UTF-32. When JSON is written in UTF-8, JSON is 8bit compatible ([RFC2045]). When JSON is written in UTF-16 or UTF-32, JSON is binary ([RFC2045]).

Fragment identifier considerations:

> The syntax and semantics of fragment identifiers specified for +json SHOULD be as specified for "application/json". (At publication of this document, there is no fragment identification syntax defined for "application/json".)
>
> The syntax and semantics for fragment identifiers for a specific "xxx/yyy+json" SHOULD be processed as follows:
>
> For cases defined in +json, where the fragment identifier resolves per the +json rules, then process as specified in +json.
>
> > For cases defined in +json, where the fragment identifier does not resolve per the +json rules, then process as specified in "xxx/yyy+json".
> >
> > For cases not defined in +json, then process as specified in "xxx/yyy+json".

Interoperability considerations: n/a

Security considerations: See [RFC4627]

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

3.2. The +ber Structured Syntax Suffix

The ITU defined the Basic Encoding Rules (BER) transfer syntax in [ITU.X690.2008]. The suffix "+ber" MAY be used with any media type whose representation follows the BER transfer syntax. (The Expert Reviewer for media type structured syntax suffix registrations ought to be aware of the relationship between BER and DER to aid in selecting the proper suffix.) The media type structured syntax suffix registration form for +ber follows:

Name: Basic Encoding Rules (BER) transfer syntax

+suffix: +ber

References: [ITU.X690.2008]

Encoding considerations: BER is a binary encoding.

Fragment identifier considerations:

At publication of this document, there is no fragment identification syntax defined for +ber.

The syntax and semantics for fragment identifiers for a specific "xxx/yyy+ber" SHOULD be processed as follows:

For cases defined in +ber, where the fragment identifier resolves per the +ber rules, then process as specified in +ber.

For cases defined in +ber, where the fragment identifier does not resolve per the +ber rules, then process as specified in "xxx/yyy+ber".

For cases not defined in +ber, then process as specified in "xxx/yyy+ber".

Interoperability considerations: n/a

Security considerations:

Each individual media type registered with a +ber suffix can have additional security considerations.

BER has a type-length-value structure, and it is easy to construct malicious content with invalid length fields that can cause buffer overrun conditions.

BER allows for arbitrary levels of nesting, which may make it possible to construct malicious content that will cause a stack overflow.

Interpreters of the BER structures should be aware of these issues and should take appropriate measures to guard against buffer overflows and stack overruns in particular and malicious content in general.

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

3.3. The +der Structured Syntax Suffix

The ITU defined the Distinguished Encoding Rules (DER) transfer syntax in [ITU.X690.2008]. The suffix "+der" MAY be used with any media type whose representation follows the DER transfer syntax. (The Expert Reviewer for media type structured syntax suffix registrations ought to be aware of the relationship between BER and DER to aid in selecting the proper suffix.) The media type structured syntax suffix registration form for +der follows:

Name: Distinguished Encoding Rules (DER) transfer syntax

+suffix: +der

References: [ITU.X690.2008]

Encoding considerations: DER is a binary encoding.

Fragment identifier considerations:

At publication of this document, there is no fragment identification syntax defined for +der.

The syntax and semantics for fragment identifiers for a specific "xxx/yyy+der" SHOULD be processed as follows:

For cases defined in +der, where the fragment identifier resolves per the +der rules, then process as specified in +der.

For cases defined in +der, where the fragment identifier does not resolve per the +der rules, then process as specified in "xxx/yyy+der".

For cases not defined in +der, then process as specified in "xxx/yyy+der".

Interoperability considerations: n/a

Security considerations:

Each individual media type registered with a +der suffix can have additional security considerations.

DER has a type-length-value structure, and it is easy to construct malicious content with invalid length fields that can cause buffer overrun conditions.

DER allows for arbitrary levels of nesting, which may make it possible to construct malicious content that will cause a stack overflow.

Interpreters of the DER structures should be aware of these issues and should take appropriate measures to guard against buffer overflows and stack overruns in particular and malicious content in general.

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

3.4. The +fastinfoset Structured Syntax Suffix

The ITU defined the Fast Infoset document format as a binary representation of the XML Information Set in [ITU.X891.2005]. These documents further define the "application/fastinfoset" media type. The suffix "+fastinfoset" MAY be used with any media type whose representation follows that established for "application/fastinfoset". The media type structured syntax suffix registration form follows:

Name: Fast Infoset document format

+suffix: +fastinfoset

References: [ITU.X891.2005]

Encoding considerations:

Fast Infoset is a binary encoding. The binary, quoted-printable, and base64 content-transfer-encodings are suitable for use with Fast Infoset.

Fragment identifier considerations:

The syntax and semantics of fragment identifiers specified for +fastinfoset SHOULD be as specified for "application/fastinfoset". (At publication of this document, there is no fragment identification syntax defined for "application/fastinfoset".)

The syntax and semantics for fragment identifiers for a specific "xxx/ yyy+fastinfoset" SHOULD be processed as follows:

For cases defined in +fastinfoset, where the fragment identifier resolves per the +fastinfoset rules, then process as specified in +fastinfoset.

For cases defined in +fastinfoset, where the fragment identifier does not resolve per the +fastinfoset rules, then process as specified in "xxx/yyy+fastinfoset".

For cases not defined in +fastinfoset, then process as specified in "xxx/ yyy+fastinfoset".

Interoperability considerations: n/a

Security considerations:

There are no security considerations inherent in Fast Infoset. Each individual media type registered with a +fastinfoset suffix can have additional security considerations.

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

3.5. The +wbxml Structured Syntax Suffix

The Wireless Application Protocol (WAP) Forum has defined the WAP Binary XML (WBXML) document format as a binary representation of XML in [WBXML]. This document further defines the "application/vnd.wap.wbxml" media type. The suffix "+wbxml" MAY be used with any media type whose representation follows that established for "application/vnd.wap.wbxml". The media type structured syntax suffix registration form follows:

Name: WAP Binary XML (WBXML) document format

+suffix: +wbxml

References: [WBXML]

Encoding considerations: WBXML is a binary encoding.

Fragment identifier considerations:

The syntax and semantics of fragment identifiers specified for +wbxml SHOULD be as specified for "application/vnd.wap.wbxml". (At publication of this document, there is no fragment identification syntax defined for "application/vnd.wap.wbxml".)

The syntax and semantics for fragment identifiers for a specific "xxx/yyy+wbxml" SHOULD be processed as follows:

For cases defined in +wbxml, where the fragment identifier resolves per the +wbxml rules, then process as specified in +wbxml.

For cases defined in +wbxml, where the fragment identifier does not resolve per the +wbxml rules, then process as specified in "xxx/yyy+wbxml".

For cases not defined in +wbxml, then process as specified in "xxx/yyy+wbxml".

Interoperability considerations: n/a

Security considerations:

There are no security considerations inherent in WBXML. Each individual media type registered with a +wbxml suffix can have additional security considerations.

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

3.6. The +zip Structured Syntax Suffix

The ZIP format is a public domain, cross-platform, interoperable file storage and transfer format, originally defined by PKWARE, Inc.; it supports compression and encryption and is used as the underlying representation by a variety of file formats. The media type "application/zip" has been registered for such files. The suffix "+zip" MAY be used with any media type whose representation follows that established for "application/zip". The media type structured syntax suffix registration form follows:

Name: ZIP file storage and transfer format

+suffix: +zip

References: [ZIP]

Encoding considerations: ZIP is a binary encoding.

Fragment identifier considerations:

The syntax and semantics of fragment identifiers specified for +zip SHOULD be as specified for "application/zip". (At publication of this document, there is no fragment identification syntax defined for "application/zip".)

The syntax and semantics for fragment identifiers for a specific "xxx/yyy+zip" SHOULD be processed as follows:

For cases defined in +zip, where the fragment identifier resolves per the +zip rules, then process as specified in +zip.

For cases defined in +zip, where the fragment identifier does not resolve per the +zip rules, then process as specified in "xxx/yyy+zip".

For cases not defined in +zip, then process as specified in "xxx/yyy+zip".

Interoperability considerations: n/a

Security considerations:

IP files support two forms of encryption: Strong Encryption and AES 128-bit, 192-bit, and 256-bit encryption; see the specification for further details. Each individual media type registered with a +zip suffix can have additional security considerations.

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller: The Apps Area Working Group. IESG has change control over this registration.

4. IANA Considerations

See the media type structured syntax suffix registration forms in Sections 3.1 - 3.6.

4.1. The +xml Structured Syntax Suffix

The following structured syntax suffix registration for "+xml" shall be used to reflect the information found in [RFC3023], with the addition of fragment identifier considerations. (Note that [RFC3023] is in the process of being updated by [XML-MEDIATYPES].)

Name: Extensible Markup Language (XML)

+suffix: +xml

References: [RFC3023]

Encoding considerations:

Per [RFC3023], XML is allowed to be represented using both 7-bit and 8-bit encodings. When XML is written in UTF-8, XML is 8bit compatible ([RFC2045]). When XML is written in UTF-16 or UTF-32, XML is binary ([RFC2045]).

Fragment identifier considerations:

The syntax and semantics of fragment identifiers specified for +xml SHOULD be as specified for "application/xml". (At publication of this document, the fragment identification syntax considerations for "application/xml" are defined in [RFC3023], Sections 5 and 7.)

The syntax and semantics for fragment identifiers for a specific "xxx/yyy+xml" SHOULD be processed as follows:

For cases defined in +xml, where the fragment identifier resolves per the +xml rules, then process as specified in +xml.

For cases defined in +xml, where the fragment identifier does not resolve per the +xml rules, then process as specified in "xxx/yyy+xml".

For cases not defined in +xml, then process as specified in "xxx/yyy+xml".

Interoperability considerations: See [RFC3023].

Security considerations: See [RFC3023]

Contact: Apps Area Working Group (apps-discuss@ietf.org)

Author/Change controller:

The Apps Area Working Group. IESG has change control over this registration.

5. Security Considerations

See the Security Considerations sections found in the media type structured syntax suffix registration forms from Sections 3 and 4.

When updating a +<suffix> registration, care should be taken to review all previously-registered xxx/yyy+<suffix> media types as to whether they might be affected by the updated +<suffix> registration. Because the generic fragment identifier processing rules take precedence over media-type-specific rules, introducing new or changing existing definitions may break the existing registrations of specific media types, as well as particular implementations of applications that process affected media types. Such changes can introduce interoperability and security issues.

When updating the fragment identifier processing rules for a specific xxx/yyy+<suffix> media type, care should be taken to review the generic fragment identifier processing rules for the +<suffix> registration and not introduce any conflicts. Because the generic fragment identifier processing rules take precedence over media-type-specific rules, such conflicting processing requirements should be ignored by an implementation, but such conflicts can introduce interoperability and security issues.

Note that [FRAGID-BP] provides additional advice to designers of fragment identifier rules for media type suffixes and specific media types.

6. References

6.1. Normative References

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[ITU.X690.2008] International Telecommunications Union, "Recommendation ITU-T X.690 | ISO/IEC 8825-1 (2008), ASN.1 encoding rules: Specification of basic encoding Rules (BER), Canonical encoding rules (CER) and Distinguished encoding rules (DER)", ITU-T Recommendation X.690, November 2008.

[ITU.X891.2005] International Telecommunications Union, "Recommendation ITU-T X.891 | ISO/IEC 24824-1 (2007), Generic applications of ASN.1: Fast infoset", ITU-T Recommendation X.891, May 2005.

[WBXML] Open Mobile Alliance, "Binary XML Content Format Specification", OMA Wireless Access Protocol WAP-192-WBXML-20010725-a, July 2001.

[ZIP] PKWARE, Inc., "APPNOTE.TXT - .ZIP File Format Specification", PKWARE .ZIP File Format Specification - Version 6.3.2, September 2007.

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

6.2. Informative References

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

[FRAGID-BP]
Tennison, J., "Best Practices for Fragment Identifiers and Media Type Definitions", July 2012, <http://www.w3.org/TR/fragid-best-practices/>.

[XML-MEDIATYPES]
Lilley, C., Makoto, M., Melnikov, A., and H. Thompson, "XML Media Types", Work in Progress, November 2012.

Authors' Addresses

Tony Hansen
AT&T Laboratories
200 Laurel Ave. South
Middletown, NJ 07748
USA

EMail: tony+sss@maillennium.att.com

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Alexey.Melnikov@isode.com