

[Draft] Minutes of the Seattle Meeting of
ISO/IEC JTC 1/SC 34/WG4, 2015-02-24/26

Rex Jaeschke (rex@RexJaeschke.com)

2015-03-06

1. Opening remarks

The meeting started at 09:50 on 2015-02-24. The convener, Murata-san, welcomed everyone to the 22nd face-to-face meeting of WG4.

2. Roll call of delegates

The following members were present during part or all of the meeting:

Name	Affiliation	Employer/Sponsor
Makoto Murata	WG4 Convener, JP	International University of Japan
Rex Jaeschke	Ecma, Project Editor	Consultant
Caroline Arms	Ecma	Library of Congress
John Haug	Ecma, US	Microsoft
Chris Rae	Ecma	Microsoft
Rich McLain	Ecma	Microsoft
Francis Cave	GB	Francis Cave Digital Publishing
Alex Brown	GB	Griffin Brown Digital Publishing Ltd.
Gareth Horton	GB	Datawatch

Present were 9 people, from 3 NBs and 1 liaison.

3. Adoption of the agenda

The agenda (SC 34 N 2141) was adopted as published, with the addition to “Other Business” of a discussion of the possibility of replacing the direct reference to PKWare’s Appnote with the newly adopted IS 21320-1 produced by WG8 (formerly WG1).

4. Administration

Approval of Previous Meeting Minutes [WG4 N 0302]

The draft minutes were approved, as circulated.

Outstanding Action Items

- None

Report from the WG4 Secretariat

The following NBs and liaisons have registered delegates to WG4: BR, CA, CH, CI, CN, CZ, DE, DK, Ecma, FI, FR, GB, IN, IT, JP, KR, NL, NO, OASIS, PL, US, W3C, XML Guild, and ZA. All requests for additions, deletions, and changes to the delegate list should be sent to the WG4 Secretariat (rex@RexJaeschke.com).

The WG4 email list is e-SC34-WG4@ecma-international.org. The document repository is now at <http://isotc.iso.org/livelink/livelink?func=ll&objid=8912947&objaction=ndocslst>.

Note: New documents are no longer being posted to the Japan-hosted website. Only the LiveLink site will be updated. Members must get themselves added to the LiveLink Global Directory through their National Body or Liaison Organization.

5. Revising Part 2 (Open Packaging Conventions)

Media Type vs. Content Type

2015-02-21 Makoto Murata “Media Type .vs Content Type in newer HTTP 1.1 RFCs”:

RFC [2616](#) (HTTP 1.1) was obsoleted by RFCs [7230](#), [7231](#), [7232](#), [7233](#), [7234](#), [7235](#). How do these RFC use media types and content types?

First, I searched for "content type" and "content-type" in a case-insensitive manner. The former never appears, while the latter appears 51 times. But the latter always appears as "Content-Type".

Note that "Content-Type" is a field name. In other words, "content type" as a noun phrase is never used.

Second, I searched for "media type" and "media-type" in a case-insensitive manner. The former appears 98 times, while the latter appears 11 times. The former is used as a noun phrase.

Some of the occurrences of "media-type" are occurrences as a non-terminal symbol.

I thus conclude that HTTP 1.1 now uses "media type" rather than "content type". The only exception is the "Content-Type" field.

2015-02-21 Makoto Murata "My proposals: content type and media types":

First, which term?

Proposal: Use "media type" except for existing element/attribute names.

Second, which RFC should we reference?

Proposal: RFC 7231 rather than RFC 2616. Certainly,

RFC 7231 depends on other RFCs even for the definition of media types, but we do not have to worry about that.

Third, which regular expression?

Proposal:

Here is a relevant part of `opc-contentTypes.xsd`.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE xs:schema
[
<!ENTITY X "[\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\&quot;\/\[\]\?=\{\}\s\t]]">
```

```
<!ENTITY Y "[\p{IsBasicLatin}-[\p{Cc}&#127;\&quot;]]|[\p{IsLatin-1Supplement}\t]">
]>
```

```
<xs:schema xmlns="http://schemas.openxmlformats.org/package/2006/content-types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.openxmlformats.org/package/2006/content-types"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  blockDefault="#all">
...
  <xs:simpleType name="ST_ContentType">
    <xs:restriction base="xs:string">
      <xs:pattern
        value="$X+/$X+(\s*;\s*($X+((($X+)|(&quot;(&Y;|(\[\p{IsBasicLatin}\p{IsLatin-1Supplement}))*&quot;)))))*"/>
      </xs:restriction>
    </xs:simpleType>
...
</xs:schema>
...
```

Notes taken by Caroline during the meeting:

Caroline agrees with Murata-san's proposal that we should reference RFC 7231.

Beginning of subclause with definition from 7231 == 3.1.1.1. Media Type

HTTP uses Internet media types [RFC2046] in the Content-Type (Section 3.1.1.5) and Accept (Section 5.3.2) header fields in order to provide open and extensible data typing and type negotiation. Media types define both a data format and various processing models: how to process that data in accordance with each context in which it is received.

```
media-type = type "/" subtype *( OWS ";" OWS parameter )
```

```
type      = token
```

```
subtype   = token
```

The type/subtype MAY be followed by parameters in the form of name=value pairs.

```
parameter = token "=" ( token / quoted-string )
```

Caroline also agreed that all instances of "content type" that unambiguously mean an instance of this syntactical structure should be replaced by "media type". Note: the definition in 7231 is different in detail from that currently in Part 2. And the 2nd sentence in 8.2.3 definitely needs fixing.

Caroline also agreed that instances of "content type" that unambiguously refer to the ContentTypes stream, including the ContentType attribute per se (as opposed to a value for the attribute) should remain as such.

However, when she started to try to make changes in clause 8, she ran into an issue not addressed directly in Murata-san's proposal. The specification is structured as a somewhat abstract package model presented in clause 8, general guidelines on mapping the package model to a physical model (sub-clause 9.2) and a specific mapping to a ZIP archive (sub-clause 9.3). See, in particular, tables 8-1, 9-1, and 9-2.

"Part name" and "Part content type," are components of the package model. Both correspond to attributes in the ContentTypes stream.

Example: <Override PartName="/word/settings.xml"

ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.settings+xml"/>

Caroline asked the group as a whole to come to an agreement over whether to change the name of the "Part content type" component of the abstract package model. Having different names is more consistent with the idea of mapping from abstract model to physical package. But it looks as though media types are the only permitted physical realization of content types based on 8.2.3.

Related facts/issues:

- Mapping for the component of the package model currently called "content type" is via the ContentTypes stream (as a ZIP item if the physical package is a ZIP archive) and the ContentType attribute. See tables 9-1 and 9-2.

N 0303 – ISO/IEC JTC 1/SC 34/WG4 Minutes of the Seattle Meeting of 2015-02-24/26

- The package model (clause 8 in WD1, clause 9 in published Part 2) is abstract in that it does not assume a ZIP archive as the container. However, it is quite physical/concrete in some other ways, including that the only typology for parts is Internet media types as defined in an RFC. This is made clear in 9.1.2 in published Part 2, which becomes 8.2.3 in WD1.

HOWEVER

- Subclause 9.2.3.2 has the following text:

9.2.3.2 Identifying the Part Content Type

The package implementer shall define a format mapping with a mechanism for associating content types with parts. [M2.3]

Some physical package formats have a native mechanism for representing content types. [Example: The content type header in MIME. end example] For such packages, the package implementer should use the native mechanism to map the content type for a part. [S2.1]

For all other physical package formats, the package implementer should include a specially named XML stream in the package, called the Content Types stream. [S2.2]

The Content Types stream shall not be mapped to a part by the package implementer. [M2.1] This stream is therefore not URI-addressable.

However, it can be interleaved in the physical package using the same mechanisms used for interleaving parts.

The first sentence of the second paragraph of 9.2.3.2 could be read to suggest that there may be representations for content types that do not use media-type syntax per RFC 7231. That would certainly conflict with sub-clause 8.2.3.

After some discussion, the group agreed that 8.2.3 is correct and 9.2.3.2 is misleading.

[Aside by Caroline re fixing 9.2.3.2: I think what can be different is the mechanism for representing the association of content types (media types) with parts, not the mechanism for representing content types.]

Based on the agreement w.r.t 8.2.3 vs. 9.2.3.2, the group agreed that changing the name of the "part content type" component of the package model to "part media type" seemed to do no harm. [Note from Caroline:

N 0303 – ISO/IEC JTC 1/SC 34/WG4 Minutes of the Seattle Meeting of 2015-02-24/26

A corollary of this would be that section headings that are tied to the package model components would also change. This was not explicitly mentioned, but I believe it was understood.]

The first paragraph of clause 7 has a use of "content types" that is definitely intended as much more general than media-types. There may be other cases where that is the case. In this case, we could switch to "content categories" to avoid confusion.

It was agreed that making changes from "content type" to "media type" should be done more carefully than simple find/replace. We may generate circularities that need to be fixed or find other instances where "content type" is used in a very general sense.

2015-02-26 John Haug:

Re: the ST_ContentType regex: I am nearly done with a lengthy explanatory break-it-down tutorial document that shows the derivation step by step! Thanks much to Murata-san for all the initial investigation and to him and Francis for talking through it on the screen yesterday at (painful) length.

The short version is that the huge 6-line regex in Part 2 is a literal translation of RFC 2616's definition of media-type into an XSD pattern. I have that part done and am working on the differences between RFC 2616 and RFC 7231 (and friends). I think it's reasonable to change the Part 2 normative reference to 7231 (and friends by reference from 7231) since it has obsoleted 2616. But we ought to understand and discuss the differences before making a concrete decision on that.

2015-02-27 John Haug:

Below is what I came up with, which should fully explain the issue and be step-by-step enough to make it easier to ensure there are no typos/errors.

Pursuant to the revision, we will need to decide:

- Whether we are OK with the differences introduced by RFC 7231
- Whether we should rewrite the XSD regex pattern by translating the RFC 7231 media-type definition (as Part 2 originally did with RFC 2616)
- Whether we should then simplify the regex in some partial or extreme way, within the limits of what XSD and RNG allow

0. Introduction

This follows on to Murata-san's e-mails in the threads "My proposals: content type and media ypest" (last from 21 Feb 2015) and "Which RFC(s) for media type should we refer to?" (last from 11 Dec 2014) analyzing the regular expression in the schema for ST_ContentType in Part 2 Annex D. It simplifies the [XML Schema regular expression](#) and compares it to RFC 2616's definition of media-type. It then compares the definition of media-type in RFC 2616 and RFC 7231, which obsoletes RFC 2616.

Tracked changes show the progression of analyzing the regex. In at least Microsoft Word, you can click on a comment to see it highlight only the segment of BNF it relates to.

John Haug, 27 Feb 2015

1. Part 2 schema regex simplification

1.1 Original Part 2 schema regex pattern (ST_ContentType)

```
(((((\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\&quot;\/\[\]\?=\{\}\s\t]]+)))/(((\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\&quot;\/\[\]\?=\{\}\s\t]]+))((\s+)*;(\s+)*(((\p{IsBasicLat
in}-[\p{Cc}&#127;\(\)&lt;&gt;@,;:\&quot;\/\[\]\?=\{\}\s\t]]+))=(((((\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\&quot;\/\[\]\?=\{\}\s\t]]+))|(&quot;(([\p{IsLatin-
1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r])|(\s+))|(\[\p{IsBasicLatin}]))*&quot;))))*))
```

1.2 Replace same chunks with X (not using correct XSD entity reference notation for simplicity)

$$\frac{\left(\left(\left(\left[\text{p}\{\text{IsBasicLatin}-\right.\right.\right.\right.$$

$$\left.\left.\left[\text{p}\{\text{Cc}\}&\#127;\backslash(\backslash)\&1\text{t};\&\text{g}\text{t};@,\text{;}\backslash\&\text{q}\&\text{u}\&\text{o}\&\text{t}^{};\backslash[\backslash]\backslash?=\backslash(\backslash)\text{s}\text{t}}\right]\right]\underline{x})+\right)/\left(\left(\left([\text{p}\{\text{IsBasicLatin}-\right.\right.\right.$$

$$\left.\left.\left[\text{p}\{\text{Cc}\}&\#127;\backslash(\backslash)\&1\text{t};\&\text{g}\text{t};@,\text{;}\backslash\&\text{q}\&\text{u}\&\text{ot}^{};\backslash[\backslash]\backslash?=\backslash(\backslash)\text{s}\text{t}}\right]\right]\underline{x})+\right)(\text{s}+)^*(\text{s}+)^*\left(\left(\left([\text{p}\{\text{IsBasicLa}\right.\right.\right.$$

$$\left.\left.\text{tin}-\right]\backslash[\text{p}\{\text{Cc}\}&\#127;\backslash(\backslash)\&1\text{t};\&\text{g}\text{t};@,\text{;}\backslash\&\text{q}\&\text{u}\&\text{ot}^{};\backslash[\backslash]\backslash?=\backslash(\backslash)\text{s}\text{t}}\right]\right]\underline{x})+\right)=\left(\left(\left([\text{p}\{\text{IsBasicLatin}-\right.\right.\right.$$

$$\left.\left.\left[\text{p}\{\text{Cc}\}&\#127;\backslash(\backslash)\&1\text{t};\&\text{g}\text{t};@,\text{;}\backslash\&\text{q}\&\text{u}\&\text{ot}^{};\backslash[\backslash]\backslash?=\backslash(\backslash)\text{s}\text{t}}\right]\right]\underline{x})+\right)|\&\text{q}\&\text{u}\&\text{ot}^{};\left(\left([\text{p}\{\text{IsLatin}-\right.\right.\right.$$


```
1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|(\s+))|(\[\p{IsBasicLatin}]])*&quot;))))*)
```

1.3 Remove obvious unnecessary parentheses

```
{(((X+)))/((X+))((\s+)*;(\s+)*(((X+))=((X+))|&quot;(([\p{IsLatin-
1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|(\s+))|(\[\p{IsBasicLatin}]])*&quot;))))*)}
```

1.4 Remove other unnecessary parentheses, modifiers (+) and character specifications (\n\r covered by \p{Cc})

```
X+/X+((\s+)*;(\s+)*((X+))=(X+|&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|(\s+))|(\[\p{IsBasicLatin}]])*&quot;))))*)
```

Remove unnecessary parentheses, replace chunk with Y (ignore whitespace added for ease of grouping/reading)

```
X+/X+(\s*; \s*(X+=(
X+ | (&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-[\p{Cc}&#127;&quot;]]|\s+))Y |
{([\p{IsBasicLatin}]])*&quot;))
)))*
```

Simplify (remove added whitespace)

```
X+/X+(\s*; \s*(X+=(X+|&quot;(Y|([\p{IsBasicLatin}]])*&quot;))))*)
```

2. Definitions #1 – RFC 2616 and regex

1.4.1 RFC 2616

NOTE: This uses old [RFC 822](#)-style augmented Backus-Naur Form (not the same as [RFC 5234](#) ABNF)

```
media-type    = type "/" subtype *( ";" parameter )
type          = token
subtype       = token
```

parameter = attribute "=" value
attribute = token
value = token | quoted-string
quoted-string = (<"> *(qdtype | quoted-pair) <">)
qdtype = <any TEXT except <">>
quoted-pair = "\" CHAR
TEXT = <any OCTET except CTLs, but including LWS>
LWS = [CRLF] 1*(SP | HT) ; linear white space
CRLF = CR LF
token = 1*<any CHAR except CTLs or separators>
CHAR = <any US-ASCII character (octets 0 - 127)>
CTL = <any US-ASCII control character (octets 0 - 31) and DEL (127)>
separators = "(" | ")" | "<" | ">" | "@" | "," | ";" | ":" | "\" | <"> | "/" | "[" | "]" | "?" | "=" | "{" | "}" | SP | HT
OCTET = <any 8-bit sequence of data>
CR = <US-ASCII CR, carriage return (13)>
LF = <US-ASCII LF, linefeed (10)>
SP = <US-ASCII SP, space (32)>
HT = <US-ASCII HT, horizontal-tab (9)>
<"> = <US-ASCII double-quote mark (34)>

Commented [JH1]: \p{IsBasicLatin}

Commented [JH2]: \p{Cc}

Commented [JH3]: \p{IsBasicLatin}\p{IsLatin-1Supplement}

1.4.2 Regex

From <http://www.regular-expressions.info/unicode.html>:

\p{Cc} or \p{Control}: an ASCII 0x00–0x1F or Latin-1 0x80–0x9F control character

From [https://msdn.microsoft.com/en-us/library/20bw873z\(v=vs.110\).aspx#SupportedNamedBlocks](https://msdn.microsoft.com/en-us/library/20bw873z(v=vs.110).aspx#SupportedNamedBlocks):

Code point range	Block name	Note
0000 - 007F	IsBasicLatin	ASCII 0-127
0080 - 00FF	IsLatin-1Supplement	Extended ASCII 128-255

3. Interpretation of X and Y in Part 2 regex

X: `[\\p{IsBasicLatin}-[\\p{Cc}\\(\\)<>;@,;:\\"\\/\\[\\]\\?={\\}\\s\\t}]`

- English approximation: Any single ASCII character except controlchars DEL () < > @ , ; : \ " / [] ? = { } SPACE TAB

X+ is the same as RFC 2616's token

Y: `([\\p{IsLatin-1Supplement}\\p{IsBasicLatin}-[\\p{Cc}"]]|\\s+)`

- English approximation: Any single extended ASCII character including linear whitespace except controlchars DEL "

Y is the same as RFC 2616's qdtext

4. Comparison of simplified Part 2 schema regex and RFC 2616 media-type

media-type = type "/" subtype *(";" parameter)

= token "/" token *(";" attribute "=" value)

= token "/" token *(";" token "=" (token | quoted-string))

= token "/" token *(";" token "=" (token | (<"> *(qdtext | quoted-pair) <">)))

= token "/" token *(";" token "=" (token | (<"> *(qdtext | "\\ CHAR" <">)))

`X+ / X+ (\\s* ; \\s* (X+ = (X+ | (" (M | \\[\\p{IsBasicLatin}]*) " ;)))) *`

5. Definitions #2 – RFC 7231

1.5 RFC 7231

NOTE: This uses [RFC 5234](#) Augmented Backus-Naur Form

Commented [JH4]: CHAR

Commented [JH5]: CTL

Commented [JH6]: separators

Commented [JH7]: type

Commented [JH8]: subtype

Commented [JH9]: attribute

Commented [JH10]: value

Commented [JH11]: type

Commented [JH12]: subtype

Commented [JH13]: OPC allows whitespace around the ; which is different from RFC 2616

\\s allows [\\t\\r\\n\\f] plus possibly vertical tab and possibly Unicode "separators"

RFC 7231 allows only [\\t]

Commented [JH14]: attribute

Commented [JH15]: token

Commented [JH16]: qdtext

Commented [JH17]: quoted-pair

Commented [JH18]: quoted-string

Commented [JH19]: value

```

media-type    = type "/" subtype *( OWS ";" OWS parameter )
type          = token
subtype       = token
parameter     = token "=" ( token / quoted-string )
quoted-string = DQUOTE *( qdtext / quoted-pair ) DQUOTE
qdtext       = HTAB / SP / %x21 / %x23-5B / %x5D-7E / obs-text
quoted-pair  = "\" ( HTAB / SP / VCHAR / obs-text )
token        = 1*tchar
tchar        = "!" / "#" / "$" / "%" / "&" / "'" / "*" / "+" / "-" / "." / "^" / "_" / "`" /
/ "|" / "~" / DIGIT / ALPHA
              ; any VCHAR, except delimiters
VCHAR        = %x21-7E          ; visible (printing) characters, RFC 5234 Appendix B.1
DIGIT        = %x30-39          ; RFC 5234 Appendix B.1
ALPHA        = %x41-5A / %x61-7A ; A-Z / a-z
obs-text     = %x80-FF
OWS          = *( SP / HTAB )   ; optional whitespace, RFC 7230 Section 3.2.3
SP           = %x20             ; RFC 5234 Appendix B.1
HTAB         = %x09             ; horizontal tab, RFC 5234 Appendix B.1

```

6. Differences between RFC 2616 and RFC 7231

1.6 media-type

RFC 2616 disallows whitespace around the semi-colon preceding a parameter. RFC 7231 allows any number of SP and/or HTAB.

1.7 token

No differences

1.8 qdtext

RFC 2616 includes LF (octet 10 / %x0A), CR (octet 13 / %x0D), \ (octet 92 / %x5C). RFC 7231 disallows these characters.

1.9 quoted-pair

RFC 2616 allows any character in the standard ASCII range (octets 0-127). RFC 7231 disallows the range octets 0-31 except for octet 9 (HTAB).

2015-02-28 Francis Cave:

Good job! I just have one niggle, which is with your use of \s in the definitions of both X and Y. The problem is that \s includes \t \n and \r, all of which are in \p{Cc}. The meaning of \s is not SPACE, but 'white space', where this includes all Unicode space characters (i.e. including U+0020 and U+00A0, but also presumably some other space characters), and also includes control characters TAB, CR and LF.

In X this isn't so critical, because you're excluding \s, but that means that there is redundancy in the expression, because \t \n and \r are already excluded by excluding \p{Cc}.

In Y the problem is more serious, because you are including \s+ as an alternative choice to the rest of the expression. Effectively this allows \t \n and \r in Y expressions that are white space only.

I suspect that the only white space character that should be allowed in Y, is U+0020, i.e. the regular SPACE character. This would be the same as SP in the ABNF in RFC 7231.

Note that U+00A0 is in the Latin-1 supplement. I'm not sure whether this character should be explicitly excluded from X. It can presumably be included in Y, as this in a quoted string.

Here is a list of Unicode spaces: <https://www.cs.tut.fi/~jkorpela/chars/spaces.html>. As this isn't an official list, I cannot be certain that this is accurate. My assumption is that \s includes all these.

I am assuming that \p{Cc} includes control characters in the Unicode range U+0080 to U+009F, as well as the control characters in the basic ASCII range.

2015-03-01 Makoto Murata:

Nice work!

I think that

("(((\p{IsLatin-1Supplement}\p{IsBasicLatin}-[\p{Cc}"]]\s+)(\\p{IsBasicLatin}})*")

cannot be simplified to

("(Y | \\p{IsBasicLatin}})*")

Rather, it should become

("(Y|(\\p{IsBasicLatin}})*")

XAdES

There were several email threads on this topic; see subjects “XAdES elements in OFF-CRYPTO of Microsoft” and “OPC WD1 - XMLDSIG/XAdES proposed changes”.

We glanced over the §12 editorial rewrites in John’s document distributed on 18 Feb but didn’t go into detailed review. We spent time discussing the proposed items for the XAdES subclause and examined some details of the proposed text with respect to the XAdES 1.4.1 text. Mainly, additional questions were raised that need to be investigated. John’s draft was updated with these comments and distributed to WG 4 on 27 Feb. National Bodies are requested to review this document and provide appropriate comments within WG 4.

On Thursday, we held a teleconference with Juan Carlos Cruellas from Universidad Politecnica de Catalunya and the ETSI ESI committee working on the next version of XAdES and related digital signatures. He gave an overview of the ongoing wholesale reorganization of the CAAdES, XAdES, PAdES and ASiC standards. The new XAdES documents, EN 319 132 parts 1 and 2, will be EN (European standards) rather than TS (ETSI technical specifications). ESI is rewriting the levels of XAdES into a different set of “baseline” and “extended” levels compared to TS 101 903. He noted key technical differences between the current TS and future EN include that XMLDSIG 1.1 deprecated the ds:X509IssuerSerial element [the problem was that the serial number is defined as an integer in the XML Schema, and a number of XML Schema validating tools ‘may not support integer types with decimal data exceeding 18 decimal digits’], and that the new XAdES will have changes regarding assertions and extension mechanism for SignedProperties.

Juan Carlos reported that the XAdES draft will be finalized by the end of 2015 April and will hopefully be approved as EN in one year or one year and a half.

Juan and WG4 agreed that collaboration between SC34 and the ESI WG of ETSI would be very fruitful. We find that there is no official liaison between JTC1 and ESI as of now. Juan and Murata will respectively study how such liaison can be established.

6. Extensions: 30114, “Extensions of Office Open XML File Formats”

Re Part 1, “Guidelines”, we agreed to use John’s document (N 0223), “MCE Best Practices”, as a basis and to add other information about non-standard OPC parts.

Action: Chris will write “Guidelines for which extensions mechanisms to use”.

Re Part 2, “Character Repertoire Checking”, we agreed that Murata-san should produce WD2.

Overnight, Murata-san produced the following skeleton for 30114-1:

§2. MCE

Use John's draft

§3. Foreign OPC parts

§3.1 General

Media types

Relationship types

§3.2 Processing

Case 1: an application program is unaware of a foreign OPC part

The application program does not handle the foreign part, but should preserve it.

Case 2: an application program is aware of a foreign OPC part

The application program may handle the foreign part and may preserve it.

§4. Comparison and Guidelines

MCE ignorables and ACBs

- Easy to use anywhere in XML parts
- Old application programs throw MCE ignorables and ACBs.

MCE application-defined extension elements

- Usable only at predefined locations in XML parts
- Old application programs preserve application-defined extension elements

Foreign OPC parts

- Detached from the original part
- Old application programs preserve foreign OPC parts

7. Defect Reports

The public, online DR log is now at

<https://onedrive.live.com/?cid=c8ba0861dc5e4adc&sc=documents&sa=501765342&id=C8BA0861DC5E4ADC%21105>. Access individual DRs via the hyperlinks contained within the spreadsheet's left-most column.

DR 13-0014 "PML: omissions and inconsistencies in the specification of attributes"

There was considerable discussion and wordsmithing. Chris has made great progress, and will continue working on the proposed resolution.

DR 14-0008 "SML: Specifying a Range in a Separate Workbook"

We revisited the discussion from the previous teleconference and confirmed that the information was available; it just took some digging to find it. Closed without action.

DR 14-0010 "SML: Attribute textRotation"

In the proposed changes, all occurrences of minInclusive were changed to maxInclusive.

Agreed with Chris' proposal. Closed in COR4.

DR 14-0011 “SML: Attribute Value of “none””

Agreed with Chris’ proposal. Closed in COR4.

DR 14-0012 “SML: Cell Type ‘d’”

There was a brief discussion.

Action: Chris will write up a reply to submitter explaining why we’ve closed this DR without action.

DR 14-0013 “SML: definedName attribute localSheetId”

Agreed with Chris’ proposal. Closed in COR4.

DR 14-0014 “SML: Merging Cells”

Chris agreed that there was something to fix here. He’ll work on a proposal.

DR 14-0015 “SML: Cell Styles”

Chris will work on a proposal.

DR 14-0016 “SML: Number Formats”

Agreed with Chris’ proposal. Closed in COR4.

DR 15-0001 “DML: Text point font size”

Chris presented his proposed response. We agreed to close this without action.

DR 15-0002 “SML: Schema for GradientFill does not limit colors”

There was some discussion about this and unbounded scenarios, in general.

Action: Chris will write up a reply to submitter explaining why we’ve closed this DR without action.

8. Other Business

Considering Pointing to IS 21320-1

There was some discussion of whether to replace the reference to PKWare's appnote.text in Part 2 with a reference to the forthcoming Standard 21320-1, "Information technology -- Document Container File -- Part 1: Core", produced by WG8 (formerly WG1). It was supposed that doing this would almost certainly have no adverse technical impact, but that it would require a large amount of checking to ensure this. Given that the result would not meaningfully improve the standard, it was felt time could better be spent on higher-priority tasks.

Thanking Host

We thanked Microsoft and John Haug for hosting the meeting and a great dinner.

9. Future meetings

Face-to-Face Meetings:

- 2015-06-15/18, BSI, London, UK (Mon & Tue: WG4 all day, Wed morning: WG8, Wed afternoon: WG4, Thu morning: WG4) If WG6 needs 1+ hours for a teleconference, we'll make time for them.
- 2015-09-21/25, Beijing, CN (with other WGs, and Opening/Closing Plenaries)

We talked about the number of F2F meetings for 2016. We might be able to get by with only two, but we'll revisit this topic at the London meeting. If the 2016 SC 34 Plenary is held in Asia, we'll have at least one WG4 meeting in Europe.

Teleconferences:

**** The 90-day ballot on the Part 1 and Part 4 DCORs ends 2015-03-16 ****

- 2015-04-09, 13:00 GMT (US/PT 06:00, GB 14:00, DE/DK/FR/CZ 15:00, JP 22:00)
- 2015-05-14, 13:00 GMT (US/PT 06:00, GB 14:00, DE/DK/FR/CZ 15:00, JP 22:00)

10. Adjournment

Adjourned by unanimous consent at 11:05 on 2015-02-26.