Changes document for SyncML Meta Information

Specification version: 1.0.1

Specification date: 2001-06-15

SyncML Initiative

The following companies are Sponsors of the SyncML Initiative:

Ericsson
IBM
Lotus
Matsushita Communications Industrial Co., Ltd.
Motorola
Nokia
Openwave
Starfish Software
Symbian

Copyright Notice

1 Formatting Conventions

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

1.1 Errata Type Classifications

The errata types are classified according to the following scheme:

CLARIFICATION: Textual enhancement that provides a clearer explanation of a specification item without changing any behavior.

CORRECTION: A modification that obsoletes some items in the current published specification.

PROBLEM: A known problem for which an erratum has yet to be proposed.

2 Errata

2.1 Parent of <Last>

2.1.1 Problem

In section 5.6 of the Meta Info specification, under “Parent Elements” the “MetInf” element is listed, but it is not correct.

2.1.2 Solution

The correct Parent for Last is “Anchor”. Change the “Parent Elements” to “Anchor”.

2.1.3 Other specifications/erratas affected

None.

2.2 Definition of MUST, SHOULD, MAY (RFC 2119)

2.2.1 Problem

The current definition is unclear about how to interpret a receiving element when the “Static Conformance Requirements” column defines an element as MAY.

In almost every document we have a reference to www.ietf.org and in chapter “Static Conformance Requirements” we have:

In these tables, optional features are specified by a "MAY", mandatory features are specified by a "MUST" and recommended features are specified by a "SHOULD".

2.2.2 Solution

Change the reference to RFC2119 and include the MAY definition from the RFC under the chapter “Static Conformance Requirements”:

“An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality.”

2.2.3 Other specifications/erratas affected

None.

2.3 Parent of <Next>

2.3.1 Problem

In section 5.11 of the Meta Info specification, under “Parent Elements” the “MetInf” element is listed, but it is not correct.

2.3.2 Solution

The correct Parent for Next is “Anchor”. Change the “Parent Elements” to “Anchor”.

2.3.3 Other specifications/erratas affected

None.

2.4 MetaInf Ref Typos

2.4.1 Problem

In section 9 of the Meta Info specification, the word Protocol in the SyncML Protocol reference is misspelled without the ‘c’.

In section 5.10 of the Meta Info specification, the example lists **`<MetInf xmlns:mi='syncml:metinf'>,`** where it should be: **`<MetInf xmlns='syncml:metinf'>`**

In section 5.12 the contraction “it’s” (it is) is used, which should be the possessive form of “it” - which is “its”.

In section 5.12 the example spells NextNonce as NextNounce, two places.

In section 7.1 the word mapped is spelled as “maped”.

2.4.2 Solution

Correct the spelling in 9 from “Protool” to “Protocol”.

Correct the example in 5.10 to “**`<MetInf xmlns='syncml:metinf'>`**”

Correct the spelling in 5.12 to “its”.

Correct the example in 5.12 from “**`<NextNounce”`** to “**`<NextNonce”`** and “**`</NextNounce>`**” to “**`</NextNonce>`**”.

Correct the spelling in 7.1 from “maped” to “mapped”.

2.4.3 Other specifications/erratas affected

None.

3 Enhancements

3.1 Device Management Formats

3.1.1 Problem

The device management protocol needs new value types added to the format element.

3.1.2 Solution

Change the definition of Format

Eg:

Restrictions:

“The value of this element SHOULD BE one of b64, chr, int, null or xml.”

to

“The value of this element SHOULD BE one of b64, chr, int, null, xml, bool, node or bin. If the value is bool, the format of the content is either true or false. If the value is node, the content represents an interior object in the management tree. If the value is bin, the format of the content is binary data.”

3.1.3 Other specifications/erratas affected

None.

3.2 Large Object Support

3.2.1 Problem

A single SyncML message is limited in size by the underlying PDU of the transport it uses. SyncML provides no mechanism to split data payload across multiple messages. When working over a wireless link using a transport such as WSP this imposes a limitation that is likely to have real world impact on the size of objects that can be synchronised.

3.2.2 Solution

syncml_metinf_v101_20010615.doc, add new element

MaxObjSize

Usage: Specifies the maximum size in bytes of a data object that the device is able to receive.

Parent Elements: MetInf

Restrictions:
The element type appears in the Meta element of a SyncML request to specify the maximum size, in bytes, of the largest object it is capable of receiving in any subsequent response messages. This element type value is applicable for the remainder of the synchronization session.

The element type value is the text string representation of the maximum, decimal byte size without leading zeroes.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example: Device that can receive a maximum object of 10K bytes

```
<MaxObjSize>10240</MaxObjSize>
```

syncml_metinf_v101_20010530.dtd, add new element

```
<!ELEMENT MaxObjSize (#PCDATA)>
```

3.2.3 Other specifications/erratas affected

syncml_protocol_v101_20010615.doc

syncml_represent_v101_20010615.doc

syncml_represent_v101_20010530.dtd

syncml_metinf_v101_20010615.doc

syncml_metinf_v101_20010530.dtd

4 References

[RFC 2119] Key words for use in RFCs to Indicate Requirement Levels, IETF.