Wireless Application Environment Defined Media Type Specification

Version 15-May-2001

Wireless Application Protocol
WAP-237-WAEMT-20010515-a

A list of errata and updates to this document is available from the WAP Forum™ Web site, http://www.wapforum.org/, in the form of SIN documents, which are subject to revision or removal without notice.

The WAP Forum™ assumes no responsibility for errors or omissions in this document. In no event shall the WAP Forum be liable for any special, indirect or consequential damages or any damages whatsoever arising out of or in connection with the use of this information.

WAP Forum™ members have agreed to use reasonable endeavors to disclose in a timely manner to the WAP Forum the existence of all intellectual property rights (IPR's) essential to the present document. The members do not have an obligation to conduct IPR searches. This information is publicly available to members and non-members of the WAP Forum and may be found on the "WAP IPR Declarations" list at http://www.wapforum.org/what/ipr.htm. Essential IPR is available for license on the basis set out in the schedule to the WAP Forum Application Form.

No representations or warranties (whether express or implied) are made by the WAP Forum™ or any WAP Forum member or its affiliates regarding any of the IPR's represented on this list, including but not limited to the accuracy, completeness, validity or relevance of the information or whether or not such rights are essential or non-essential.

This document is available online in PDF format at http://www.wapforum.org/.

Known problems associated with this document are published at http://www.wapforum.org/.

Comments regarding this document can be submitted to the WAP Forum™ in the manner published at http://www.wapforum.org/.

Document History	
WAP-237-WAEMT-20010515-a	Current

Contents

1. SCOPE 4

2. REFERENCES 5

2.1. NORMATIVE REFERENCES 5

2.2. INFORMATIVE REFERENCES 5

3. TERMINOLOGY AND CONVENTIONS 6

3.1. CONVENTIONS 6

3.2. DEFINITIONS 6

3.3. ABBREVIATIONS 6

4. WIRELESS BITMAP FORMAT 7

4.1. WBMP TYPE IDENTIFIERS 7

4.2. WBMP SYNTAX 8

4.3. DATA TYPES 9

4.3.1. Multi-byte Integer Format 9

4.4. HEADER DATA STRUCTURE 9

4.4.1. Header Formats 9

4.4.2. FixHeaderField 10

4.4.3. Extension Field Type 11 10

4.5. SPECIFICATION OF WELL-DEFINED WBMP TYPES 11

4.5.1. WBMP Type 0: B/W, Uncompressed Bitmap 11

APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) 12

APPENDIX B. CHANGE HISTORY (INFORMATIVE) 13

1. Scope

Wireless Application Protocol (WAP) is a result of continuous work to define an industry wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications. The wireless market is growing very quickly and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation and fast/flexible service creation, WAP defines a set of protocols in transport, session and application layers.

Wireless Application Environment (WAE) is part of the WAP Forum's effort to specify an application framework for wireless terminals such as mobile phones, pagers, and PDAs. The framework extends and leverages other WAP technologies, including WTP and WSP, as well as other Internet technologies such as XML, URLs, scripting, and various media types. The effort enables operators, manufacturers, and content developers to meet the challenges in building advanced and differentiating services and implementations in a fast and flexible manner.

This specification defines content formats that are optimised for wireless environment. Currently, this specification includes following format:

- Wireless BitMaP (WBMP)

For a general feature of the overall WAE architecture, please refer to [WAESpec]

2. References

2.1. Normative References

[CREQ] "Specification of WAP Conformance Requirements". WAP Forum™. WAP-221-CREQ. URL:http//www.wapforum.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels". S. Bradner. March 1997. URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997. URL:http://www.ietf.org/rfc/rfc2234.txt

[HTTP/1.1] "Hypertext Transfer Protocol -- HTTP/1.1", RFC2616, R. Fielding et al., June 1999. URL:http://www.ietf.org/rfc/rfc2616.txt

[WSP] "Wireless Session Protocol", WAP Forum™. WAP-230-WSP. URL:http://www.wapforum.org/

2.2. Informative References

[WAESpec] "Wireless Application Environment Specification Version 2.0". WAP Forum™. WAP-236-WAESpec. URL:http//www.wapforum.org/

[WML] "Wireless Markup Language", WAP Forum™. WAP-238-WML2. URL:http://www.wapforum.org/

[WMLScript] "WMLScript Language Specification", WAP Forum™. WAP-193-WMLS. URL:http://www.wapforum.org/

3. Terminology and Conventions

3.1. 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 [RFC2119].

All sections and appendixes, except “Scope”, are normative, unless they are explicitly indicated to be informative.

3.2. Definitions

Content - synonym for data objects.

Conatent Format - actual representation of content.

User Agent - a user agent is any software or device that interprets WML[WML], WMLScript[WMLScript] or other content. This may include textual browsers, voice browsers, search engines, etc.

3.3. Abbreviations

ABNF	Augmented BNF
BNF	Backus-Naur Form
B/W	Black and White
HTTP	HyperText Transfer Protocol [HTTP/1.1]
LSB	Least Significant Bit
MSB	Most Significant Bit
PDA	Personal Digital Assistant
RFC	Request For Comments
WAP	Wireless Application Protocol
WAE	Wireless Application Environment
WBMP	Wireless BitMaP
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WWW	World Wide Web

4. Wireless Bitmap Format

The WBMP format enables graphical information to be sent to a variety of mobile terminals. The WBMP format is terminal independent and describes only graphical information.

4.1. WBMP Type Identifiers

The WBMP format is configured according to a type field value (TypeField below), which maps to all relevant image encoding information, such as:

- Pixel organisation and encoding
- Palette organisation and encoding
- Compression characteristics
- Animation encoding

For each TypeField value, all relevant image characteristics are fully specified as part of this document. Currently, a simple compact, monochrome or black and white(B/W) image format is defined within the WBMP type space. Although black and white is referred to throughout this specification it means the two states of pixel off and on respectively regardless of the display technology being used.

See Table 4-1 for the type identifiers.

Image Type Identifier, mb_u_int32	*Image Format description*
0	B/W, no compression

Table 4-1: WBMP Image Type identifier assignments

The MIME media type for WBMP is defined as follows:

```
image/vnd.wap.wbmp; level=0
```

The required `level` parameter refers to the WBMP types described in Table 4-1. In the current version of the specification, only `level=0` is defined. Any system, which supports WBMP, MUST support WBMP type 0 (WAEMT-WBMP-C-001). The specification of WBMP level 0 is given in section 4.5.1.

When initialising a session with a WAP server using WSP[WSP], or when invoking an HTTP[HTTP/1.1] request, the User Agent MUST report all supported WBMP types at least using HTTP/WSP `Accept` request header if it wants to advertise its supported image types(WAEMT-WBMP-C-002); e.g.,

```
Accept: image/vnd.wap.wbmp; level=0
```

When a WBMP image is conveyed over HTTP/WSP, the `Content-Type` header shall be used to indicate the MIME media type of the content; e.g.,

```
Content-Type: image/vnd.wap.wbmp; level=0
```

Although the server is required to indicate the correct MIME media type using the `Content-Type` header, the User Agent is recommended to decide the actual media type of the content by examining the actual data format. The actual data format has precedence over the media type indicated in the `Content-Type` header. For example, if the HTTP/WSP header indicates:

```
Content-Type: image/vnd.wap.wbmp; level=1
```

while TypeField value of the WBMP header is 0, then the User Agent may assume the WBMP level 0 as the media type of that content.

4.2. WBMP syntax

The following is a BNF-like description of the tokenised structure. The description uses the conventions established in [RFC2234].

```
WBMP                   = Header Image-data
Header                 = TypeField FixHeaderField [ExtFields] Width Height
TypeField              = mb_u_int32
                       ; Type of image which is defined in Section 4.1
FixHeaderField         = byte
                       ; Octet which is defined in Table 4-3

ExtFields             = *ExtFieldType00 / ExtFieldType01
                      / ExtFieldType10 / *ExtFieldType11
ExtFieldType00         = byte     ; Octet which is defined in Section 4.3
ExtFieldType01         = byte     ; Octet which is defined in Section 4.3
ExtFieldType10         = byte     ; Octet which is defined in Section 4.3
ExtFieldType11         = ParameterHeader ParameterIdentifier ParameterValue

ParameterHeader        = byte     ; Octet which is defined in Table 4-4
ParameterIdentifier= 1*8CHAR
                       ; Parameter identifier (US-ASCII string),
                       ; length ≤ 8 bytes defined in ParameterHeader
ParameterValue         = 1*16(ALPHA / DIGIT)
                       ; Parameter value (alphanumeric string), length ≤ 16
                       ; bytes defined in ParameterHeader
Width                  = mb_u_int32
                       ; Horizontal width of the bitmap in pixels
Height                 = mb_u_int32
                       ; Vertical height of the bitmap in pixels
Image-data             = Main-image 0*15Animated-image
                       ; There can be 0 to 15 animated images
Main-image             = *byte
                       ; Bitmap formed according to image data structure
                       ; specified by the TypeField
Animated-image         = *byte
                       ; Bitmap formed according to image data structure
                       ; description below'
```

4.3. Data Types

The following data types are used in the specification of the WBMP format.

Data Type	*Definition*
byte	8 bits of opaque data
mb_u_int32	32 bit unsigned integer, encoded in multi-byte integer format.

Table 4-2: Data Types used in WBMP format

4.3.1. Multi-byte Integer Format

A multi-byte integer consists of a series of octets, where the most significant bit (MSB) is the continuation flag, and the remaining seven bits are a scalar value. The continuation flag is used to indicate that an octet is not the end of the multi-byte sequence. A single integer value is encoded into a sequence of N octets. The first N-1 octets have the continuation flag set to a value of one (1). The final octet in the series has a continuation flag value of zero. The remaining seven bits in each octet are encoded in a big-endian order, i.e., most significant bit first. The octets are arranged in a big-endian order, i.e., the most significant seven bits are transmitted first. In the situation where the initial octet has less than seven bits of value, all unused bits must be set to zero (0).

For example, the number 0x87A5 (1000 0111 1010 0101) is encoded in three octets as follows:

1	0000010	1	0001111	0	0100101

Figure 4-1: Multi Byte Integer

The unsigned integer MUST be encoded in the smallest encoding possible. In other words, the encoded value MUST NOT start with an octet with the value 0x80.

4.4. Header Data Structure

4.4.1. Header Formats

```
Header              = TypeField FixHeaderField [ExtFields] Width Height
TypeField           = mb_u_int32
FixHeaderField      = byte
ExtFields           = *ExtFieldType00 / ExtFieldType01
                    / ExtFieldType10 / *ExtFieldType11
Width               = mb_u_int32
Height              = mb_u_int32
```

The header field contains an image type identifier of multi-byte integer (`TypeField`), an octet of general header information (`FixHeaderFiled`), zero or more extension header fields (`ExtField`), a multi-byte width field (`Width`) and a multi-byte height field (`Height`).

The extension headers may be of type binary *00* through binary *11*, defined as follows.

- Type 00 indicates a multi-byte bitfield used to specify additional header information. The first bit is set if a type 00 extension header is set if more data follows. The other bits are reserved for future use.
- Type 01 - reserved for future use.

- Type 10 - reserved for future use.
- Type 11 indicates a sequence of parameter/value pairs. These can be used for optimisations and special purpose extensions, e.g., animation image format. The “parameter size” tells the length (1-8 bytes) of the following parameter name. The “value size” gives the length (1-16 bytes) of the following parameter value. The concatenation flag indicates whether another parameter/value pair will follow after reading the specified bytes of data.

The actual organisation of the image data depends on the image type.

4.4.2. FixHeaderField

```
FixHeaderField     = byte
```

The FixHeaderField is a single byte and structured as follows:

Bit(s)	***Description***
7 (MSB)	Indicates whether extension header field(s) (i.e., ExtFields) follows the FixHeaderField. If this bit is zero, the header contains no extension header fields. If this bit is one, the FixHeaderField is followed immediately by one or more extension header field.
6-5	Indicates extension header type, i.e., type 00, 01, 10, or 11. See section 4.4.1.
4-0	Reserved

Table 4-3: FixHeaderField description

4.4.3. Extension Field Type 11

```
ExtFieldType11     = ParameterHeader ParameterIdentifier ParameterValue
ParameterHeader    = byte
ParameterIdentifier= 1*8CHAR
ParameterValue     = 1*16(ALPHA / DIGIT)
```

An extension header of type 11 consists of a sequence of parameter/value pairs. The `ParameterHeader` field (i.e., the first octet) in each extension header of type 11 is structured as follows:

Bit(s)	***Description***
7 (MSB)	Indicates whether extension header field(s) follows the current extension header field. If this bit is zero (0), the current extension header field is the last extension header field. If this bit is one (1), the current extension header field is followed immediately by one or more extension header fields.
6-4	Indicates size of Parameter Identifier in bytes. (e.g., if these bits are *110*, the size of the Parameter Identifier is 6 bytes)
3-0	Indicates size of Parameter Value in bytes. (e.g., if these bits are *1000*, the size of the Parameter Value is 8 bytes)

Table 4-4: Extension Field Type 11

4.5. Specification of well-defined WBMP Types

4.5.1. WBMP Type 0: B/W, Uncompressed Bitmap

WBMP type 0 has the following characteristics:

- No compression
- Colour: one bit represents a pixel (white=1, black=0).
- Depth: 1 bit deep (monochrome)
- The high bit of each byte is the left-most pixel of the byte.
- The first row in the data is the upper row of the image.

The WBMP header encoding is shown in the Tables Table 4-5 and Table 4-6.

Field Name	***Value***	***Description***
TypeField	0	Indicates WBMP type 0: B/W, Uncompressed Bitmap
FixHeaderField	(an octet)	See Table 4-6: FixHeaderField description.
ExtHeaderField(s)	-	Extension headers is not presented.
Width	(any value)	A mb_u_int32 value indicates the width of the image in pixels.
Height	(any value)	A mb_u_int32 value indicates the height of the image in pixels.

Table 4-5: Length of Header Parts

The FixHeaderField of the type 0 WBMP header is as follows:

Bit(s)	***Value***	***Description***
7 (MSB)	0	Indicates that the header contains no extension header fields.
6-5	0	Extension Header Type. Ignored.
4-0	0	Reserved.

Table 4-6: FixHeaderField description

Extension headers MUST NOT be presented in this format.

The WBMP image data is organised in pixel rows, which are represented by a sequence of octets. One bit represents one pixel intensity with value white=1 and value black=0. In the situation where the row length is not divisible by 8 the encoding of the next row must start at the beginning of the next octet, and all unused bits must be set to zero (0). The data bits are encoded in a big-endian order; i.e., most significant bit first. The octets are arranged in a big-endian order, i.e., the most significant octet is transmitted first. The most significant bit in a row represents the intensity of the left most pixel. The first row in the image data is the top row of the image.

The WBMP image can have at most 15 animated images following the main image. It is User Agent dependent how those animated images are processed.

Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [CREQ].

Item	Function	Reference	Status	Requirement
WAEMT-WBMP-C-001	Support for WBMP (B/W, Uncompressed Bitmap) image/vnd.wap.wbmp; level=0	4.1	M	
WAEMT-WBMP-C-002	Informs supported WBMP types using Accept header	4.1	M	

Appendix B. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	15-May-2001		The initial version of this document.