Wireless Profiled HTTP

Version 29-Mar-2001

Wireless Application Protocol
WAP-229-HTTP-20010329-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.

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-229-HTTP-20010710-a	current

Contents

1. Scope

The 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, reaching new customers and providing new services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of open, extensible protocols and content formats as a basis for interoperable implementations.

The objectives of the WAP Forum are:

- To bring Internet content and advanced data services to digital cellular phones and other wireless terminals.
- To create a global wireless protocol specification that will work across differing wireless network technologies.
- To enable the creation of content and applications that scale across a very wide range of bearer networks and device types.
- To embrace and extend existing standards and technology wherever appropriate.

As part of the evolution of the WAP Specifications, WPG has decided to converge the WAP stack with key internet protocols defined by the IETF. This involves the inclusion of the following protocols to the WAP stack

- HTTP [RFC2616]
- TLS Profile and Tunnelling [WAPTLS]
- Wireless Profiled-TCP [WTCP]

The scope of this document is the Transfer Services Layer which provides for the structured transfer of rich information between network elements [WAPARCH]. The scope of this document is to define a protocol independent service access point to the Transfer Services layer, specifically Hypermedia Transfer, for use by the upper-level application layer of WAP and to define the profile of HTTP 1.1 [RFC2616], i.e. the features and functionality, to be provided by the layer. It is not the intention of this document to define support for other transfer layer services such as Streaming and Messaging.

2. References

2.1. Normative References

[CREQ] "Specification of WAP Conformance Requirements". WAP Forum™. WAP-221-CREQ-20010425-a. http//www.wapforum.org/

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

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

[WAPTLS] "WAP TLS Profile and Tunneling", WAP-199-WTLS-20000218-a.WAP Forum. http://www.wapforum.org/

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1", R.Fielding, J.Gettys, …, June 1999. http://www.ietf.org/rfc/rfc2616.txt

[RFC2817] "Upgrading to TLS Within HTTP/1.1," RFC2817, R. Khare, S. Lawrence, May 2000. http://www.ietf.org/rfc/rfc2817.txt

2.2. Informative References

[WAPARCH] "WAP Architecture". WAP Forum™. WAP-210-WAPArch-20001130-p. URL:http//www.wapforum.org/

[WSP] "Wireless Session Protocol". WAP-203-WSP-20000504-a.WAP Forum. http://www.wapforum.org/

[WTP] "Wireless Transaction Protocol Specification", WAP-201-WTP-20000219-a.WAP Forum. http://www.wapforum.org/

[WTCP] "Wireless Profiled TCP Specification", WAP-225-TCP-20010321-p. http://www.wapforum.org/

[ISO7498] "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model", ISO/IEC 7498-1:1994.

[Performance] "Network Performance Effects of HTTP 1.1, CSS1 and PNG", Henrik Frystyk Nielsen, Jim Gettys et al. June 1997. http://www.w3.org/TR/NOTE-pipelining

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” and “Architectural Overview” are normative, unless they are explicitly indicated to be informative.

3.2. Definitions

Author – an author is a person or program that writes or generates WML, WMLScript or other content.

Client – a device (or application) that initiates a request for a connection with a server.

Content – subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request.

Content Encoding – when used as a verb, content encoding indicates the act of converting content from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process.

Content Format – actual representation of content.

Device – a network entity that is capable of sending and receiving packets of information and has a unique device address. A device can act as both a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

Entity - An entity is the information transferred as the payload of a request or response. An entity consists of meta-information in the form of entity-header fields and content in the form of an entity-body.

Header - A header contains meta-information. An entity-header contains meta-information about a particular request, response or entity body (content).

Method - Method is the type of client request as defined by HTTP/1.1 (eg, Get, Post, etc.). Extended Methods are also permitted.

Origin Server – the server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

Server – a device (or application) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client.

Terminal – a device providing the user with user agent capabilities, including the ability to request and receive information. Also called a mobile terminal or mobile station.

User – a user is a person who interacts with a user agent to view, hear, or otherwise use a resource.

User Agent – a user agent is any software or device that interprets WML, WMLScript, WTAI or other resources. This may include textual browsers, voice browsers, search engines, etc.

Wireless Profiled HTTP – Detailed normative references to [RFC2616] to standardize its use by devices in WAP.

3.3. Abbreviations

A-SAP	Application – Service Access Point
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol [RFC2616]
ISO	International Standards Organisation
RFC	Request For Comments
SAP	Service Access Point
S-SAP	Session Services – Service Access Point
SEC-SAP	Security Services – Service Access Point
TCP	Transport Control Protocol
T-SAP	Transport Services – Service Access Point
TS-SAP	Transfer Services – Service Access Point
TLS	Transport Layer Security
W3C	World Wide Web Consortium
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
W-HTTP	Wireless Profiled HTTP
WWW	World-Wide Web

4. Wireless Profiled HTTP Architectural Overview

This section is informative.

4.1. Reference Model

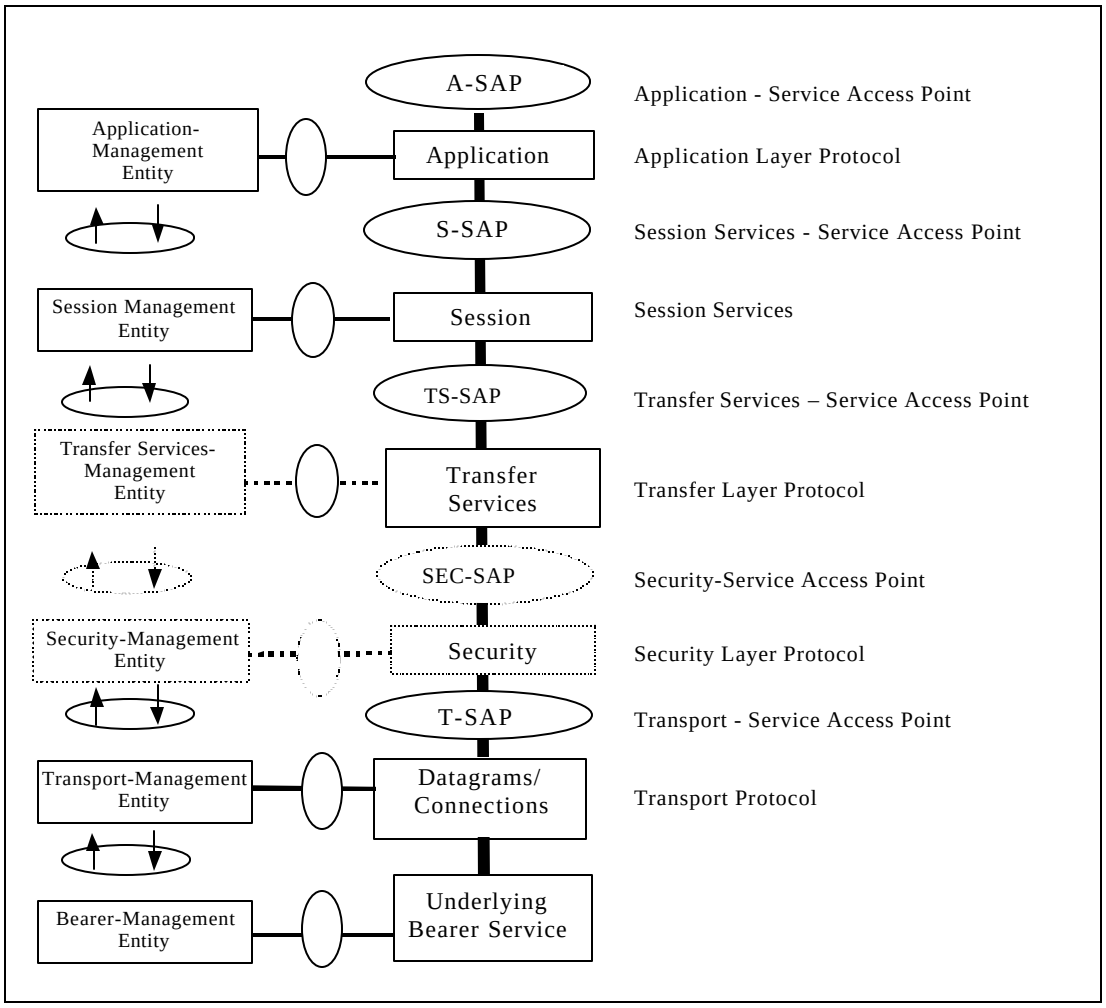

Figure 1. Wireless Application Protocol Next Generation Reference Model

A model of layering the protocols in WAP is illustrated in Figure 1. WAP protocols and their functions are layered in a style resembling that of the ISO OSI Reference Model [ISO7498]. Layer Management Entities handle protocol initialisation, configuration and error conditions (such as loss of connectivity due to the mobile station roaming out of coverage) that are not handled by the protocol itself.

The Transfer Services to be defined by this document are :

- Hypermedia Transfer – The hypermedia transfer services provides for the transfer of self-describing hypermedia resources. The combination of WSP (Wireless Session Protocol) [WSP] and WTP (Wireless Transaction Protocol) [WTP] provide the hypermedia transfer service over secure and non-secure datagram transports. The HTTP (Hypertext Transfer Protocol) [RFC2616] provides the hypermedia transfer service over secure and non-secure connection-oriented transports.

Other transfer services which are beyond the scope of this document are:

- Streaming – The streaming services provide a means for transferring isochronous data such as audio and video.
- Message Transfer – The message transfer services provide the means to transfer asynchronous multimedia messages such as email or instant messages.

The transfer services layer uses a connection-orientated transport service. Other protocols may be defined to use the datagram based transport service. Security is assumed to be an optional layer below the transfer layer. The security layer preserves the transport service interfaces. The transfer and application management entities are assumed to provide the additional support that is required to establish security contexts and secure connections.

Security support is not provided by the W-HTTP protocol directly. In this regard, the security layer is modular. W-HTTP itself does not require a security layer; however, applications that use W-HTTP may require it.

In addition to the direct access, the architecture also includes the use of the proxies between a WAP client and an origin server. The following diagrams illustrates the role of HTTP in the architecture.

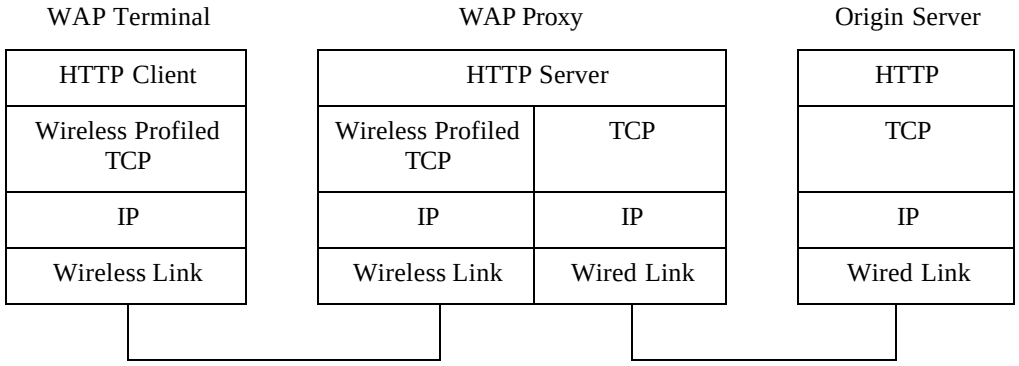

Figure 2: Wireless Profiled HTTP With WAP Proxy

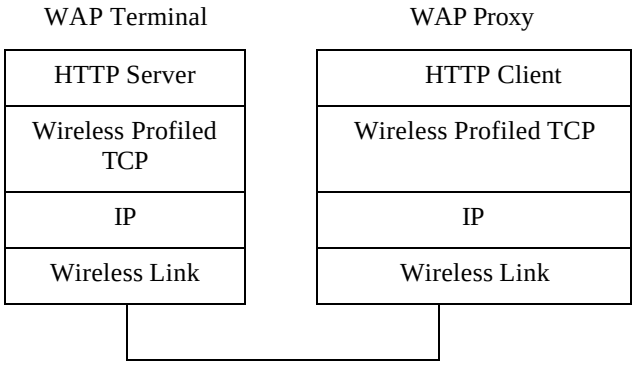

Figure 3: Wireless Profiled HTTP for WAP Push

4.2. Wireless Profiled HTTP Features

The core of the wireless profiled HTTP specification is the HTTP specification [RFC2616]. Elements and descriptions have been taken from this specification and declared as mandatory or optional in this specification. The basic model of interaction between the WAP Terminal and WAP Proxy/WAP Server is a HTTP request/response.

The WAP Terminal must be capable of interacting with WAP HTTP Proxies and Origin Servers. This transfer layer provides a service access point which may be used in both the 'pull' and 'push' data transfer models. Pull is achieved using the request/response mechanism from HTTP/1.1. Push functionality is achieved by changing the role of the WAP Terminal and considering it as an HTTP server. By doing so 'push' can be modelled as a request/response towards the WAP Terminal.

W-HTTP supports message body compression of responses, making the most efficient use of airtime.

W-HTTP supports the establishment of a Tunnel using the CONNECT method which in turn enables solutions for such problems as end to end security.

5. Wireless Profiled HTTP Protocol Operation

This section is normative.

This section describes the protocol conformance required to realise the Wireless Profiled HTTP specification. The description has been divided into four areas :

- WAP Terminal
  - HTTP Client
  - HTTP Server
- WAP Proxy
  - HTTP Client
  - HTTP Server

The HTTP Client in the WAP Proxy and the HTTP Server in the WAP Terminal are defined for use by WAP Push.

5.1. WAP Terminal HTTP Support

5.1.1. HTTP Client

The HTTP Client in the WAP Terminal MUST support the following methods as defined by HTTP [RFC2616]:

- GET
- POST

The HTTP client in the WAP Terminal MUST support the following HTTP method, if TLS is supported [WAPTLS].

- CONNECT

5.1.2. HTTP Server

If supported, the HTTP Server in the WAP Terminal MUST support the following methods as defined by HTTP [RFC2616]:

- GET
- HEAD
- POST
- OPTIONS

5.2. WAP Proxy Support

5.2.1. HTTP Client

If supported, the HTTP Client in the WAP Proxy MUST support the following methods as defined by the HTTP specification [RFC2616]:

- GET
- POST
- OPTIONS

5.2.2. HTTP Server

The HTTP Server implementation in the WAP Proxy MUST support the following methods as defined by the HTTP specification [RFC2616]:

- GET
- HEAD
- POST
- CONNECT

The HTTP Server in the WAP Proxy MAY support the following method as defined by the HTTP specification [RFC2616]:

- OPTIONS

5.3. Support for Standard HTTP Features

The methods that must be supported by the HTTP client and HTTP server on the WAP Terminal and WAP Proxy have already been identified in section 6.1 and 6.2 of this document.

5.3.1. Content Coding

In order to minimise the volume of data sent over the air the HTTP Client in the WAP Terminal MAY support content-coding and the HTTP Server in the WAP Proxy SHOULD support content coding of the message body within a HTTP response. The content coding mechanism is as specified in Section 14.11 of [RFC2616].

The WAP Proxy SHOULD support content encoding. When supported, the WAP Proxy MUST at least provide for deflate coding as specified in [RFC1951]. The WAP Terminal negotiates for content-encoding using the standard "Accept-Encoding" request header, e.g. "Accept-Encoding:deflate".

Efficient use of compression has been investigated by the W3C [Performance], whose study provides criteria to use in deciding when compression should be applied by a WAP Proxy. The WAP Proxy SHOULD NOT encode the message body of a response if a content coding has already been applied to the message body, indicated by the header "Content-Encoding" or if the content-type indicates that the data is already optimised e.g. "application/vnd.wap.wmlc".

5.4. Extended Features

5.4.1. Establishing a Tunnel with CONNECT

The HTTP Server in the WAP Proxy MUST support the establishment of a Tunnel using the CONNECT method as described in Section 5 of [RFC2817]. Once active in this role, the WAP Proxy is not considered a party to the HTTP communication.

The HTTP Client in the WAP Terminal MUST support the establishment of a tunnel using the CONNECT method if TLS is supported [WAPTLS].

Appendix A. Static Conformance Requirements

These conformance requirements have been assembled in compliance with the WAP Specification of conformance requirements [CREQ].

A.1 WAP-Terminal

Item	Functionality	Reference	Status	Requirement
HTTP-C-C001	Support for HTTP Client	6.1.1	M	HTTP-C-C002 AND HTTP-C-C003 AND TCP:MCF[WTCP]
HTTP-C-C002	Support for TLS	6.4.1	O	HTTP-C-C005 AND TLS:MCF[WAPTLS]
HTTP-C-S001	Support for HTTP Server	6.1.2	O	HTTP-C-S002 AND HTTP-C-S003 AND HTTP-C-S004 AND HTTP-C-S005 AND TCP:MSF [WTCP]
HTTP-C-C003	Support for GET Method	6.1.1, Section 9.3 [RFC2616]	O	
HTTP-C-C004	Support for POST Method	6.1.1, Section 9.5 [RFC2616]	O	
HTTP-C-C005	Support for CONNECT Method	6.1.1, 6.4.1 [RFC2817]	O	TLS-C-200 AND TLS-C-201 AND TLS-C-202 AND TLS-C-203 [WAPTLS]
HTTP-C-C006	Support for ‘deflate’ content decoding	6.3.1, Section 3.5 [RFC2616], [RFC1951].	O	
HTTP-C-S002	Support for GET	6.1.2, Section 9.3 [RFC2616]	O	
HTTP-C-S003	Support for POST	6.1.2, Section 9.5 [RFC2616]	O	
HTTP-C-S004	Support for HEAD	6.1.2, Section 9.4 [RFC2616]	O	
HTTP-C-S005	Support for OPTIONS	6.1.2, Section 9.2 [RFC2616]	O	
HTTP-C-S006	Support for ‘deflate’ content encoding	6.3.1, Section 3.5 [RFC2616], [RFC1951].	O	

A.2 WAP Proxy

Item	Functionality	Reference	Status	Requirement
HTTP-S-S001	Support for HTTP Server	6.2.2	M	HTTP-S-S002 AND HTTP-S-S003 AND HTTP-S-S004 AND HTTP-S-S006 AND TCP:MSF[WTCP]
HTTP-S-C001	Support for HTTP Client	6.2.1	O	HTTP-S-C002 AND HTTP-S-C003 AND HTTP-S-C004 AND TCP:MCF[WTCP]
HTTP-S-C002	Support for GET Method	6.2.1, Section 9.3 [RFC2616]	O	
HTTP-S-C003	Support for POST Method	6.2.1, Section 9.5 [RFC2616]	O	
HTTP-S-C004	Support for OPTIONS Method	6.2.1 Section 9.2 [RFC2616]	O	
HTTP-S-S002	Support for GET Method	6.2.2, Section 9.3 [RFC2616]	O	
HTTP-S-S003	Support for POST Method	6.2.2, Section 9.5 [RFC2616]	O	
HTTP-S-S004	Support for HEAD Method	6.2.2, Section 9.4 [RFC2616]	O	
HTTP-S-S005	Support for OPTIONS Method	6.2.2, Section 9.2 [RFC2616]	O	
HTTP-S-S006	Support for CONNECT Method	6.2.2, 6.4.1, Section 5 [RFC2817]	O	
HTTP-S-S007	Support for content encoding using ‘deflate’	6.3.1, Section 3.5 [RFC2616],[RFC 1951]	O	

Appendix B. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	29 March 2001		Frozen version of document
Class 0	16 May 2001		Update of document for new WAP Template
Class 0	10 July 2001		Updated after Approval