WAG UAProf

Version 20-Oct-2001

Wireless Application Protocol
WAP-248-UAPROF-20011020-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.

© 2001, Wireless Application Protocol Forum, Ltd. All Rights Reserved. Terms and conditions of use are available from the WAP Forum™ Web site (http://www.wapforum.org/what/copyright.htm).

© 2001, Wireless Application Forum, Ltd. All rights reserved.

Terms and conditions of use are available from the WAP Forum™ Web site at http://www.wapforum.org/what/copyright.htm.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. You may not use this document in any other manner without the prior written permission of the WAP Forum™. The WAP Forum authorises you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services offered by you.

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-248-UAProf-20011020-a	Current
WAP-248_001-UAProf-20010915-a	SIN
WAP-248_002-UAProf-20010915-a	SIN
WAP-248_003-UAProf-20011020-a	SIN

Contents

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

1. Scope

The *User Agent Profile* specification is concerned with capturing classes of device capabilities and preference information. These classes include (but are not restricted to) the hardware and software characteristics of the device as well as information about the network to which the device is connected. The user agent profile contains information used for *content formatting purposes*. A user agent profile is distinct from a *user preference profile* that would contain application-specific information about the user for content *selection* purposes. For example, a user preference profile might designate whether the user is interested in receiving sports scores and, if so, the particular teams. The specification of user preference profiles is beyond the scope of this document.

In accordance with Composite Capability/Preference Profiles (CC/PP) [CCPP], the user agent profile schema is defined using an RDF schema and vocabulary [RDF-Schema]. This document defines the structure of this schema in terms of the class definitions and semantics of attributes for devices adhering to the WAP standard. Extensibility guidelines are also offered to assist in extending the schema with new components.

Regarding the user agent profile, this document assumes that:

- The information contained within the profile is provided on behalf of the user who will be receiving the content contained in the associated WSP or HTTP response. Profiles may also be used by the Push Proxy Gateway (PPG).
- The CC/PP repository storing the profile is secure, meaning that it does not permit unauthorized modification to stored profile information. The functionality associated with this repository may be implemented in a WAP gateway or intermediate proxy or as a separate standalone element in the network.
- WSP/HTTP headers are generally not encrypted. Because this specification does not define any security mechanisms, care will be taken when including user-confidential information[1] in the profile unless the profile is transmitted over an end-to-end secure channel (e.g. TLS/WTLS to the origin server).
- An implicit chain of trust exists between the client and origin server. The integrity of the profile is maintained (in other words, not compromised) as it is transmitted through or cached within the network. It is assumed that the network elements that contribute property descriptions to the profile are trusted. Network elements will not assemble a "history" about users by tracking the deltas in their profile over time.
- To ensure the integrity of the profile, lower-level mechanisms, such as TLS or WTLS, must be used.

Use of the profile by the origin server is intended to optimize the content to the characteristics of the device, and the users' preferences. If the profile be discarded, corrupted, or otherwise inaccessible, the origin server should still provide content to the client.

[1] The definition of "*confidential*" varies among different users, cultures, and service providers. It is therefore impossible to define a profile schema that meets the security needs of all users while still conveying any useful information.

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

1.1. Relationship to Other Standards

This specification builds upon and coexists with numerous WAP and Internet standards. These relationships are summarized below:

- Composite Capability/Preferences Profiles (CC/PP) [CCPP]: This specification defines the Capabilities and Preference Information (CPI) structure according to the structure mandated by the CC/PP specification.
- Resource Description Framework (RDF) [RDF]: The CC/PP specification uses the RDF syntax to represent the CPI. In designing or extending the schema, a schema designer must be familiar with the RDF concepts. Mimetype for RDF is **application/rdf+xml** as specified in [RFC3023].
- Wireless Session Protocol (WSP) [WSP]: One of the mechanisms used to transport CPI over wireless networks is within WSP headers
- WAP Binary XML (WBXML) [WBXML]: When the CPI is transmitted over the WSP protocol, this specification mandates that it must be encoded according to the WBXML specification.
- CC/PP Exchange Protocol over HTTP [CCPPex]: Previous versions of UAProf [UAProf1] allowed for the transmission of CPI using the CC/PP Exchange Protocol over HTTP which, in turn, defines headers in HTTP 1.1 [HTTP] with the HTTP Extension Framework [HTTPext].
- HTTP [HTTP]: Optionally, the CPI can be transmitted over Wireless Profiled HTTP [W-HTTP] from the mobile terminal to the origin server. In this case, proprietary headers specified in this specification is used to convey the information.
- WAP Push Access Protocol (PAP) [WAP-PAP]: This protocol is used by push origin servers to retrieve the CPI from the Push Proxy Gateway (PPG).

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

2. References

2.1. Normative References

[CCPP] “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies”, G. Klyne, F. Reynolds, C. Woodrow, H. Ohto.
URL: http://www.w3.org/TR/CCPP-struct-vocab

[CCPPex] “CC/PP exchange protocol based on HTTP Extension Framework”, H. Ohto, J. Hjelm, June 1999.
URL: http://www.w3.org/TR/NOTE-CCPPexchange

[CREQ] “Specification of WAP Conformance Requirements”. WAP Forum™. WAP-221-CREQ-20000915-a.
URL: http//www.wapforum.org/

[HTTP] “Hypertext Transfer Protocol HTTP/1.1”, RFC2616, R. Fielding, et al., June 1999.
URL: ftp://ftp.isi.edu/in-notes/rfc2616.txt

[HTTPext] “An HTTP Extension Framework”, RFC 2774, H. Nielsen, P. Leach, and S. Lawrence, February 2000.
URL: ftp://ftp.isi.edu/in-notes/rfc2774.txt

[RDF] “Resource Description Framework (RDF) Model and Syntax Specification”, World Wide Web Consortium Recommendation, O. Lassila, R. Swick, February 1999.
URL: http://www.w3.org/TR/REC-rdf-syntax/

[RDF-Schema] “RDF Schema Specification”, World Wide Web Consortium, D. Brickley, R. V. Guha, March 1999.
URL: http://www.w3.org/TR/PR-rdf-schema

[RFC1321] “The MD5 Message-Digest Algorithm”,RFC 1321, R. Rivest, April 1992
URL: ftp://ftp.isi.edu/in-notes/rfc1321.txt

[RFC3066] “Tags for the Identification of Languages”, RFC 3066, H. Alvestrand, January 2001.
URL: ftp://ftp.isi.edu/in-notes/rfc3066.txt

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

[RFC2045] “Multipurpose Internet Mail Extensions (MIME) Part One”, RFC2045, N. Freed et al., November 1996.
URL: ftp://ftp.isi.edu/in-notes/rfc2045.txt

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

[RFC2396] “Uniform Resource Identifiers (URI): Generic Syntax”, RFC 2396, T. Berners-Lee, R. Fielding, and L. Masinter, August 1998.
URL: ftp://ftp.isi.edu/in-notes/rfc2396.txt

[WAE] “Wireless Application Environment Specification”, WAP Forum™, WAP-236-WAESpec.

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

URL: http://www.wapforum.org/

[WBXML]	"WAP Binary XML Content Format", WAP Forum™, WAP-240-WBXML. URL: http://www.wapforum.org/
[W-HTTP]	"WAP Wireless Profiled HTTP", WAP Forum™. WAP-229-HTTP. URL: http://www.wapforum.org/
[WSP]	"Wireless Session Protocol Specification", WAP Forum™, WAP-230-WSP. URL: http://www.wapforum.org/
[WTA]	"Wireless Telephony Application Specification", WAP Forum™, WAP-169-WTA. URL: http://www.wapforum.org/
[XML]	"Extensible Markup Language (XML) 1.0 (Second Edition)", World Wide Web Consortium Recommendation, Tim Bray et al., 6 October 2000. URL: http://www.w3.org/TR/2000/REC-xml-20001006
[XML-NS]	"Namespaces in XML", W3C Recommendation, T. Bray, D. Hollander, A. Layman, January 1999. URL: http://www.w3.org/TR/1999/REC-xml-names

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

2.2. Informative References

[BLT]	"Specification of the Bluetooth System, profiles version 1.1", Bluetooth SIG, Inc., February 2001. URL: http://www.bluetooth.com/developer/specification/Bluetooth_11_Profiles_Book.pdf
[CHTR]	"Wireless Application Group: UAPROF Drafting Committee Charter", WAP Forum™, February 2001. URL: http://www.wapforum.org/
[IANA]	"Internet Assigned Numbers Authority". URL: http://www.iana.org
[MexE]	"ETSI Special Mobile Group (SMG) 4", Mobile Station Application Execution Environment (MexE). URL: http://www.etsi.org
[PICT]	"Pictogram Specification", The WAP Forum™, WAP-213-Pictogram. URL: http://www.wapforum.org/
[P3P]	"The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", World Wide Web Consortium Recommendation, L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle. URL: http://www.w3.org/TR/P3P
[RFC2703]	"Protocol-independent Content Negotiation Framework", RFC2703, G. Klyne, September 1999. URL: ftp://ftp.isi.edu/in-notes/rfc2703.txt
[RFC3023]	"XML Media Types", RFC3023,. M. Murata, S. St.Laurent, D. Kohn, January 2001. URL: ftp://ftp.isi.edu/in-notes/rfc3023.txt
[SiRPAC]	SiRPAC Compiler and Parser. URL: http://www.w3.org/RDF/Implementations/SiRPAC
[UA-Attrs]	"Client-specific Web Services by using User Agent Attributes", Tomihisa Kamada, et. al, December 1997. URL: http://www.w3.org/TR/NOTE-agent-attributes
[UAProf1]	"User Agent Profile Specification", WAP Forum™, WAP-174-UAProf. URL: http://www.wapforum.org/
[WAPARCH]	"WAP Architecture Specification", WAP Forum™, WAP-210-WAPArch. URL: http://www.wapforum.org/
[WAP-PAP]	"Push Access Protocol Specification", WAP Forum™, WAP-247-PAP. URL: http://www.wapforum.org/
[WAP-PushArch]	"Push Architectural Overview", WAP Forum™, WAP-250-PushArch. URL: http://www.wapforum.org/
[WAP-PushOta]	"Push OTA Protocol Specification", WAP Forum™, WAP-235-PushOTA. URL: http://www.wapforum.org/
[WINA]	"WAP Interim Naming Authority", WAP Forum™.

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

URL: http://www.wapforum.org/wina

[WTAI]	"Wireless Telephony Application Interface Specification", WAP Forum™, WAP-170-WTAI. URL: http://www.wapforum.org/
[WMLStdLib]	"WMLScript Standard Libraries Specification", WAP Forum™. WAP-194-WMLSL. URL: http://www.wapforum.org/
[XHTML]	"XHTML™ 1.0: The Extensible HyperText Markup Language - A Reformulation of HTML 4 in XML 1.0", Steve Pemberton et al, 26 January 2000. URL: http://www.w3.org/TR/xhtml1/
[XMOD]	"Modularization of XHTML", World Wide Web Consortium Recommendation, Murray Altheim et al, 10 April 2001. URL: http://www.w3.org/TR/xhtml-modularization/

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

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 “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2. Definitions

Attribute: A UAPROF attribute refers to the data elements describing the CPI and is represented as an RDF property element. Each UAPROF attribute is associated with a single value or a list of values or resources.

CC/PP Repository: A server that stores CPI persistently in a form that may be referenced by and incorporated into a profile. A CC/PP repository is typically a Web server that provides CC/PP profiles in response to HTTP requests.

Component: Elements of a high level classification of the information in the CPI. For UAProf, these include *HardwarePlatform, SoftwarePlatform, NetworkCharacteristics, WAP, MExE, BrowserUA and PushComponent..*

CPI: Capabilities and Preference Information pertaining to the capabilities of the device, the operating and network environment, and user’s personal preferences for receiving content and/or resource. CPI are represented by means of a UAPROF Profile.

Gateway: Software that is capable of bridging disparate network protocols. For the purposes of this specification, “gateway” refers to protocol bridging functionality, which may exist in a stand-alone gateway or may be co-located with a proxy or origin server.

Home Location Register: A permanent database used in cellular networks. The HLR is located on the SCP (Signal Control Point) of the cellular provider of record, and is used to identify/verify a subscriber; it also contains subscriber data related to features and services.

Origin Server: Software that can respond to requests from a WAP terminal by delivering appropriate content or error messages. The origin server may receive requests via either WSP or HTTP. Application programs executing on the origin server can use UAProf to deliver content that is tailored in accordance with the CPI that can be found within the provided Profile. For the purpose of this specification, “origin server” refers to content generation capabilities, which may physically exist in a stand-alone Web server or may be co-located with a proxy or gateway.

Profile: An instance of the UAPROF schema that describes capabilities for a specific device and network configuration. A profile need not have all the attributes identified in the vocabulary/schema.

Profile repository: The profile repository is an origin server that stores profile resources. Profiles are treated as regular static or dynamic resources served using the access scheme specified as a part of the URI identifying the profile of a particular device. To increase speed, the origin server can use a cache to reduce the number of downloads from profile repositories. It is expected that profile repositories would typically be run by device manufacturers or software companies providing information about their user agents.

Property: An RDF property represents a specific aspect, characteristic, capability or relation (meta data) used to describe a resource. It is denoted using RDF property element syntax.

Proxy: Software that receives HTTP requests and forwards that request toward the origin server (possibly by way of an upstream proxy) using HTTP. The proxy receives the response from the origin server and forwards it to the requesting client. In providing its forwarding functions, the proxy may modify either the request or response or provide other value-added functions. For the purposes of this specification, “proxy” refers to request/response forwarding functionality, which may exist in a stand-alone HTTP proxy or may be co-located with a gateway or origin server.

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Resource: An object or element being referred to in a UAPROF expression is a resource. A UAPROF resource is typically identified by a URI.

Schema: An RDF schema denotes resources which constitute the particular unchanging versions of an RDF vocabulary at any point in time. It is used to provide semantic information (such as organization and relationship) about the interpretation of the statements in an RDF data model. It does not include the values associated with the attributes.

Server: Software which respond to HTTP requests. This software may reside on the mobile terminal. The requests may be for CPI or use CPI as meta data.

User: An individual or group of individuals acting as a single entity. The user is further qualified as an entity who uses a device to request content and/or resource from a server.

User agent: A program, such as a browser, running on the device that acts on a user's behalf. Users may use different user agents at different times.

Vocabulary: A collection of attributes that adequately describe the CPI. A vocabulary is associated with a schema. The vocabulary for UAProf includes attributes pertaining to the device capabilities and network characteristics.

3.3. Abbreviations

CC/PP	COMPOSITE CAPABILITY/PREFERENCES PROFILES
CC/PPEX	CC/PP EXCHANGE PROTOCOL
CC/PP-HTTP	CC/PP EXCHANGE PROTOCOL OVER HTTP
CC/PP-WSP	CC/PP EXCHANGE PROTOCOL OVER WSP
CPI	CAPABILITY AND PREFERENCE INFORMATION
HTML	HYPER-TEXT MARKUP LANGUAGE
HTTP	HYPER-TEXT TRANSFER PROTOCOL
IANA	INTERNET ASSIGNED NUMBERS AUTHORITY
OTA	OVER THE AIR I.E. IN THE RADIO NETWORK
PAP	PUSH ACCESS PROTOCOL
PPG	PUSH PROXY GATEWAY
P3P	PLATFORM FOR PRIVACY PREFERENCES PROJECT
RDF	RESOURCE DESCRIPTION FRAMEWORK
SIRPAC	SIMPLE RDF PARSER AND COMPILER
UAPROF	USER AGENT PROFILE
URI	UNIFORM RESOURCE IDENTIFIER
URL	UNIFORM RESOURCE LOCATOR
W3C	WORLD WIDE WEB CONSORTIUM
WAP	WIRELESS APPLICATION PROTOCOL
WBXML	WAP BINARY XML
W-HTTP	WIRELESS PROFILED HTTP
WML	WIRELESS MARKUP LANGUAGE
WSP	WIRELESS SESSION PROTOCOL
WTA	WIRELESS TELEPHONY APPLICATION
W-TCP	WIRELESS PROFILED TCP
WTLS	WIRELESS TLS
XHTML	EXTENSIBLE HYPERTEXT MARKUP LANGUAGE
XML	EXTENSIBLE MARKUP LANGUAGE

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

4. Introduction

This section is informative.

As WAP-enabled devices come of age, an assumption of device homogeneity is no longer universally valid. In particular, mobile devices can be expected to have an an ever-divergent range of input and output capabilities, network connectivity, and levels of scripting language support. Moreover, users may have content presentation preferences that also cannot be transferred to the server for consideration. As a result of this device heterogeneity and the limited ability of users to convey their content presentation preferences to the server, clients may receive content that they cannot store, that they cannot display, that violates the desires of the user, or that takes too long to convey over the network to the client device.

Work is ongoing in the World-Wide Web Consortium (W3C) to define mechanisms for describing and transmitting information about the capabilities of Web clients and the display preferences of Web users. The Composite Capabilities/Preferences Profile (CC/PP) specification [CCPP] defines a high-level structured framework for describing this information using the Resource Description Framework (RDF) [RDF]. CC/PP profiles are structured into named "components," each containing a collection of attribute-value pairs, or properties. Each component may optionally provide a default description block containing either a set of default values for attributes of that component or a URI that refers to a document containing those default values. Any attributes explicitly provided in the component description therefore override the default values provided in the default description block or through that URI. The CC/PP specification does not mandate a particular set of components or attributes, choosing instead to defer that definition to other standards bodies. The mechanism by which the profile is transported between the mobile terminal, WAP Gateway and origin server is defined in this specification

The User Agent Profile (UAProf) specification extends WAP 2.0 to enable the end-to-end flow of a User Agent Profile (UAProf), also referred to as ***Capability and Preference Information*** (CPI), between the WAP client, the intermediate network points, and the origin server. It seeks to interoperate seamlessly with the emerging standards for Composite Capability/Preference Profile (CC/PP) [CCPP] distribution over the Internet. It uses the CC/PP model to define a robust, extensible framework for describing and transmitting CPI about the client, user, and network that will be processing the content contained in a WSP/HTTP response. The specification defines a set of components and attributes that WAP-enabled devices may convey within the CPI.[2] This CPI may include, but is not limited to, hardware characteristics (screen size, color capabilities, image capabilities, manufacturer, etc.), software characteristics (operating system vendor and version, support for MExE, list of audio and video encoders, etc.), application/user preferences (browser manufacturer and version, markup languages and versions supported, scripting languages supported, etc.), WAP characteristics (WML script libraries, WAP version, WML deck size, etc.), and network characteristics (bearer characteristics such as latency and reliability, etc.). This specification seeks to minimize wireless bandwidth consumption by using a binary encoding for the CPI and by supporting efficient transmission and caching over WSP in a manner that allows easy interoperability with HTTP.

As a request travels over the network from the client device to the origin server, each network element may optionally add additional profile information to the transmitted CPI. These additions may provide information available solely to that particular network element. Alternatively, this information may override the capabilities exposed by the client, particularly in cases where that network element is capable of performing in-band content transformations to meet the capability requirements of the requesting client device.

Origin servers, gateways, and proxies can use the CPI to ensure that the user receives content that is particularly tailored for the environment in which it will be presented. Moreover, this specification permits the origin server to select and deliver services that are appropriate to the capabilities of the requesting client. Finally, it is expected that this specification will be used to enhance content personalization based on user preferences, and other factors, as specified by the Platform for Privacy Preferences Project (P3P) [P3P].

[2] Though a set of well-known components and attributes are defined within this specification, individual implementors are free to provide additional components and attributes with their CPI. However, most origin servers or proxies are unlikely to properly interpret those extensions unless they are standardized by another standards body.

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

5. End-to-End Architecture

This section is informative.

This section provides an overview of the end-to-end architecture.

WAP clients can connect to an origin server using the legacy WAP stack or Wireless Profiled HTTP. WSP clients may connect to servers using a WAP gateway, or directly to an origin server that provides support for that protocol. W-HTTP clients can connect to an origin server directly, or using a feature or performance enhancing proxy.

This specification provides for the end-to-end specification, delivery, and processing of composite capability information from the device. The information is collected on the client device, optionally enhanced with information provided with a particular request, optionally combined with other information available over the network, and made available to the origin server. Over the Internet, this specification assumes the use of the CC/PP [CCPP], HTTP, and optionally the CC/PP Exchange Protocol over HTTP [CCPPex], and HTTP 1.1 [HTTP] with the HTTP Extension Framework [HTTPext].

A client can connect to the origin server as described in the overall WAP Architecture specification [WAPARCH], the nodes involved in this specification are depicted in Figure 5.1. Note, that the protocol used to retrieve information stored in the profile repository is not specified in this specification.

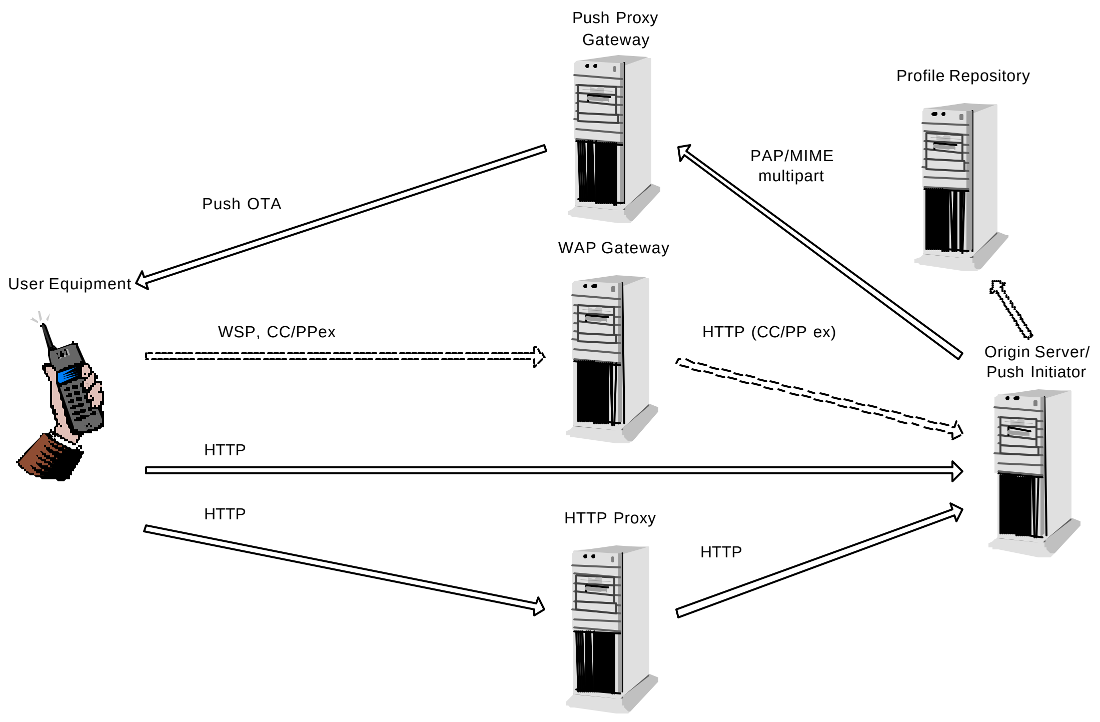


Figure 5.1:UAProf end-to-end architecture.

6. Usage Scenarios

This section and its subsections are informative.

This Section describes several scenarios in which CPI is conveyed and used to support information delivery to a WAP-enabled client. The description is divided into three sections, describing WAP/WSP and HTTP clients, and common usage scenarios.

6.1. WAP/WSP Clients

6.1.1. Opening a WSP Session and Establishing an Initial UAProf

Upon opening a WSP session with a WAP gateway, the UAProf-aware client conveys its profile information using Profile and Profile-Diff headers within the WSP **Connect** request. The values of these headers are constructed by encoding the CPI using a WBXML encoding. Upon receiving the profile, a WAP gateway that is aware of the UAProf capability responds with a Profile-Warning header value of 100 ("OK"). This header signals to the client that the CPI is being cached by the WAP gateway and will be effective for the lifetime of the session. The client device may update the CPI at any time during the session lifetime.

If the client does not receive the "OK" Profile-Warning header in the WSP **Connect** response, it assumes that the gateway does not support this UAProf specification and therefore that the CPI is not being cached by the WAP gateway. The client device may neither convey nor update the CPI during the session lifetime.

6.1.2. Updating the UAProf During an Active WSP Session

While a UAProf-aware session is established, the client may update the active UAProf at any time. To do this, the client transmits a WSP Session **Resume** message to the WAP gateway, said message containing Profile and Profile-Diff headers with the new CPI. Upon updating the cached headers, the gateway responds with a Profile-Warning header value of 100 ("OK"). All future requests issued on the WSP session will be associated with the newly cached Profile and Profile-Diff headers.

6.1.3. Resuming a Suspended WSP Session

To resume a suspended WSP session, the client initiates a standard WSP **Resume** request. The session, once resumed, retains all of the cached header state at the gateway, including the Profile and Profile-Diff headers containing the CPI.

The lifespan of a WSP session is not tied to power cycles of the device. In many situations, therefore, the client's capabilities may change significantly while a WSP session is suspended. This is the case, for example, if hardware is added or removed from the device while it is turned off. It may also occur if the WSP session is held by a smart card that may be moved to a new device while the session is suspended. In these situations, the client device must update the cached WSP Profile and Profile-Diff headers upon resuming the session. These updated headers are conveyed in the WSP **Resume** request and cached at the WAP gateway, in the same manner described in Section 6.1.2.

6.1.4. Suspending an Active WSP Session

To suspend a WSP session, the client initiates a standard WSP **Suspend** request. As long as the session is established, the WAP gateway must cache all negotiated headers, including the Profile and Profile-Diff headers associated with the WSP session. However, should the gateway choose to discard the session, it may also discard these cached headers. To resume the session, the client device follows the steps outlined in Section 6.1.3.

6.1.5. Issuing a WSP Request for Content

To request content during a UAProf-aware WSP session, the client issues a standard WSP request to the WAP gateway. The WAP gateway is responsible for forwarding this WSP request to the designated origin server (typically via HTTP).

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

In forwarding the request to the origin server, the WAP gateway must include the CPI associated with the WSP session over which the request was conveyed. The origin server receives the HTTP request (which may have been modified by one or more intermediate HTTP proxies), resolves the CPI, and generates a response along with a Profile -Warning header indicating whether the CPI was honored as the content was generated.

The HTTP response may itself be modified by intermediate HTTP proxies to better meet the needs of the requesting client. The WAP gateway forwards the content to the client device over WSP, encoding the Profile-Warning header for efficient transmission over the wireless network.

6.1.5.1. CPI Provided by the WAP Gateway

As it forwards the request, the WAP gateway may optionally add profile information, reflecting information only available to the gateway. For instance, if a network operator controls the gateway, then the gateway may provide additional network information (such as from a Home Location Register (HLR)) that is not otherwise available to the requesting client. Similarly, the WAP gateway may add information to the profile to override information provided by the requesting device/user. These overrides may, for example, reflect policies in place by the network or gateway operator.

6.1.5.2. Overriding the CPI Within a Single Request

When issuing a WSP request, the client may provide additional information to override or augment the basic CPI already cached at the WAP gateway. This additional information is only applied during the scope of the associated request, and it is not used by the gateway during subsequent requests.

To augment the profile during a request, the client includes Profile and/or Profile -Diff headers with the WSP request. As it generates an HTTP request, the WAP gateway overrides the cached WSP Profile and Profile -Diff headers with any headers provided in the WSP request. As described above in Section 6.1.5.1, the gateway may also add additional information to the forwarded profile.

6.2. Wireless Profiled HTTP Clients

The UAProf capable handset may transmit the CPI data in the x-wap-profile and x-wap-profile-diff headers with each HTTP request that is made. The user agent should also be capable of processing the x-wap-profile -warning header, which indicates whether the CPI was used when formulating the response.

6.3. Push environment

The WAP Push environment [WAP-PushArch, WAP-PAP] uses UAPROF to allow the *push-initiator* (e.g. the application generating push messages) to adapt the content of push-messages it generates based on the target device's capabilities and preferences information (CPI). Furthermore, the push-initiator (PI) can supply a constraining UAPROF setting which is matched by the Push Proxy Gateway (PPG) against the target device's CPI in determining necessary and possible content transformations respectively discarding the push-message request altogether.

6.3.1. Client Capability Query

In order to inform the PI on the target device's CPI, the PI can request these from the PPG using a client capability query (CCQ) [WAP-PAP]. Based on the target device's PAP target address, PPG retrieves the corresponding, currently valid UAPROF formatted CPI and includes this capability entity as part of the CCQ-response as described in [WAP-PAP].

The PI, being aware of the current CPI of a target device can use application specific logic to tailor the content of subsequent push-messages it generates.

6.3.2. Profile Matching

The PI can append a capability entity to an outgoing push-message which is formatted as a MIME multi-part related message.

Upon reception of the push-message's capability entity, PPG will match this profile requirement against the currently valid CPI of the target device. (If multiple target devices are addressed, the matching process is executed in turn for each individual target address.) If the required profile matches the target device profile settings, the message is forwarded. In order for such matching to occur, PPG may apply appropriate transformations to the push-message's content. However, such transformations are at the discretion of a particular PPG implementation, there is no means for the PI to control the content transformations applied at the PPG. If no match can be achieved with the relevant client capabilities, the push-message is rejected using the appropriate WAP Push status reporting [WAP-PAP].

As one particular example, consider the PI requiring the target device to support audio output. It thus will include the UAProf SoundOutputCapable hardware platform component (see Appendix A) attribute requiring it to be "True". If the PPG determines that the target device(s) does currently not support sound output, it will reject the push-message.

Matching of required properties against the currently valid CPI can only be done at the PPG once all possible or necessary content transformations have been applied. As an example, consider the PushMsgSize attribute: although the PI may indicate a value for this attribute, it is only indicative since the PI does not have knowledge of the push-message size after header compression and WML tokenization have taken place (assuming WSP is used OTA). Furthermore, PPG must validate the push-message size against the currently valid CPI and not the possibly outdated values precedently queried by the PI.

6.3.3. Push OTA

The push OTA protocol [WAP-PushOta] is designed to run on top of either WSP or HTTP. The two protocol variants are referred to as "OTA-WSP" and "OTA-HTTP" respectively. When OTA-WSP is used the profile and profile-diff headers are conveyed from the terminal to the Push Proxy Gateway (PPG) within the connect request during push session establishment. If the PPG supports UAProf a profile-warning header with status code 100 will be included in the response to indicate that the profile and profile-diff headers were correctly understood. Hence, the profile and profile-diff headers are used as request headers in this case.

On the other hand, when OTA-HTTP is utilised the PPG instead sends a registration request to the terminal to obtain its capability and preference information. This is accomplished using the HTTP OPTIONS method; the PPG sends an OPTIONS request to the terminal whereupon the terminal includes its CPI headers in the response. The CPI headers defined in [WAP-PushOta] reflects the most common capability information. The profile and profile-diff headers may also be included among the CPI headers to provide more extensive information. The headers may even replace the defined CPI headers if the PPG indicates that it supports UAProf by including a profile-warning header with warning code 200 in the request. Consequently, the profile and profile-diff headers (see Section 9) are used as response headers when OTA-HTTP is used.

6.4. Profile resolution

6.4.1. Resolving Attribute Values in the CPI

An origin server or proxy that needs to determine the correct values for CPI attributes must resolve the profile. This resolution process applies a collection of default attribute values and then applies appropriate overrides to those defaults. Because different network elements may provide additional (or overriding) profile information, the resolution process must apply this additional information to determine the final attribute values.

The User Agent Profile is constructed in three stages:

- Resolve all indirect references by retrieving URI references contained within the profile

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

- Resolve each Profile and Profile-Diff document by first applying attribute values contained in the default URI references and by second applying overriding attribute values contained within the category blocks of that Profile or Profile-Diff.
- Determine the final value of the attributes by applying the resolved attribute values from each Profile and Profile-Diff in order, with the attribute values determined by the resolution rules provided in the schema. Where no resolution rules are provided for a particular attribute in the schema, values provided in profile diffs are assumed to override values provided in previous profiles or profile diffs.

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

7. Composite Profile Segments and Attributes

This section is normative, all examples are informative. Throughout this section, sentences in Italics are also informative.

The CC/PP framework [CCPP] enables information about the capabilities and characteristics of a device and network to be communicated to web servers and gateways/proxies so that suitable content is rendered to the device. These capabilities and characteristics are referred to as attributes, and together form a vocabulary. The syntax for attributes values and - to some extent the - semantics are identified in a schema for that vocabulary. A profile is an instance of the schema and contains one or more attributes from the vocabulary. The attributes in the schema are classified into one of several components, each of which represents a distinguished set of characteristics. Components are typed and may contain a default attribute which refers to the default setting of the various component attributes. The default values are overridden by explicitly including a CC/PP attribute in a profile.

CC/PP is based on the Resource Description Framework (RDF) [RDF]. RDF is an XML application and specifies a context free grammar which well-formed RDF documents must respect. Note that not all RDF compliant documents can be described using an XML Document Type Definition (DTD); certain RDF features cannot be captured through an XML DTD. Furthermore, RDF allows various notations to convey identical meaning. For instance an XML element or an XML element attribute can be used to represent an RDF property. This specification restricts the usage of RDF in order to ease the manipulation of RDF compliant UAPROF documents.

7.1. Schema Layout

The definition of a CC/PP schema is governed by the following rules:

- The schema MUST correspond to a vocabulary which MUST be specified in conformity with the RDF [RDF] standard and for which the unique XML namespace [XML-NS] name "prf", prefixed with the XML namespace prefix "xmlns:", MUST serve as the unambiguous identifier.
- Other reserved UAProf schema XML namespace prefixes are "rdf" and "rdfs" which MUST be used to identify the RDF respectively the RDF Schema namespace.
- All XML namespace declarations used within the UAProf schema definition MUST be declared as attributes of the **rdf:RDF** root element. Namespace aliases MUST NOT be introduced within nested XML elements of the UAProf schema.
-
- The UAProf schema MUST consist of one or more CC/PP components, each describing a set of properties (or attributes) within one or more RDF description elements.
- Each component MUST be an object of RDF type *Class* with an **rdfs:label** element which is unique across the set of components introduced by the UAProf schema.
- Attribute values are typed and MUST obey their corresponding syntax. Attributes of type "Literal" MUST be interpreted in a case-sensitive manner. Leading and trailing white space MUST be discarded. Embedded linear white space is processed according to LWS rules of HTTP/1.1 [HTTP]. The syntax of attribute types is given in the UAProf schema.
- Descriptions to override the default values MAY be included in the schema's component descriptions. The final value of a profile instance attributes is resolved based on the resolution rule associated with the attribute.

7.2. User Agent Profile Components

The schema for WAP User Agent Profiles consists of description blocks for the following key components:

HardwarePlatform: A collection of properties that adequately describe the hardware characteristics of the terminal device. This includes, the type of device, model number, display size, input and output methods, etc.

SoftwarePlatform: A collection of attributes associated with the operating environment of the device. Attributes provide information on the operating system software, video and audio encoders supported by the device, and user's preference on language.

BrowserUA: A set of attributes to describe the HTML browser application

NetworkCharacteristics: Information about the network-related infrastructure and environment such as bearer information. *These attributes can influence the resulting content, due to the variation in capabilities and characteristics of various network infrastructures in terms of bandwidth and device accessibility.*

WapCharacteristics: A set of attributes pertaining to WAP capabilities supported on the device. This includes details on the capabilities and characteristics related to the WML Browser, WTA [WTA], etc.

PushCharacteristics: A set of attributes pertaining to Push specific capabilities supported by the device. This includes details on supported MIME-types, the maximum size of a push-message shipped to the device, the number of possibly buffered push-messages on the device, etc.

Additional components can be added to the schema to describe capabilities pertaining to other user agents such as an Email application or hardware extensions. See Section 7.8 for details.

7.3. Attributes

A profile attribute MUST belong to one and only one component of the schema. If an origin server application is interested in a set of attributes that spans multiple components, it MUST parse the complete profile to obtain the necessary information.

Profile attributes MUST be defined using a traditional name and value pair syntax. The first half of the name-value pair describes the attribute, and the other half provides the value itself. The RDF property descriptions MUST be unique and unambiguous in both semantics and value.

Within an RDF component description block, the attribute description MUST be either embedded lexically inline to denote the value or expressed as an RDF resource (using indirect or remote reference such as URIs) that must be resolved to obtain the full description.

Attributes with composite or multiple values MUST be described as RDF resources. For instance, the *Default* attribute points to a collection of attributes and should therefore be described as a URI resource. Similarly, an RDF container (Bag or Sequence) MUST be used to describe a list of values (ordered or unordered) associated with a given attribute. This specification constrains CC/PP to the extent that RDF containers MUST NOT contain elements that are RDF containers themselves. *For example, the InputCharSet attribute would be expressed as follows*

```
<prf:InputCharSet>
<rdf:Bag>
     <rdf:li>US-ASCII</rdf:li>
     <rdf:li>Shift_JIS</rdf:li>
</rdf:Bag>
</prf:InputCharSet>
```

The server parsing and interpreting the profile MAY carry out validation of the attribute descriptions in terms of units and value.

The value of an attribute with multiple descriptions MUST be resolved as follows:

- The description of the attribute within the Default tag MUST be resolved first
- Any other description of the attribute identified in a subsequent instantiation of the attribute MUST override the default description

- Where multiple descriptions of the attribute exist outside the default description block, the ultimate value of the attribute MUST be determined by the resolution rules for that attribute. An attribute MUST be associated with one of the following three resolution rules:

1. **Locked:** The final value is determined by the first occurrence of the attribute outside the default description block. Subsequent occurrences MUST be ignored.

2. **Override:** The final value equals the last occurrence of the attribute with a description element.

3. **Append:** The final value is a list of all the occurrences of the attribute

   For example, the *ModelNumber* attribute has a resolution semantic identified as *Locked*. This implies that the first non-default occurrence of the attribute is the final value or that subsequent instantiations of the attribute occurrence cannot change the value of that attribute. On the other hand, an attribute such as *ColorCapable* has a resolution semantic of *Override* which means that of the multiple occurrences of the attribute, only the last determines the final value. Finally, an attribute with a resolution semantic *Append* has a final value that includes all the occurrences for that attribute in the profile.

An attribute (RDF property) MUST closely represent the semantics of the capability or characteristics being represented and MUST follow the naming conventions identified in [RDF] and [RDF-Schema]. For example, the *SoundOutputCapable* attribute indicates whether or not the device supports sound output.

7.4. Profile

A profile is an instance of the schema. The following, RDF and CC/PP compliant, syntax rules govern the formation of a profile:

- The root element of the profile, labeled **rdf:RDF**, MUST be identified with an invariant *ID* attribute, set to e.g. "MyProfile".
- Profiles MUST use the XML namespace prefix "*rdf*" to refer to the RDF namespace and "*prf*" for the UAProf schema namespace. These namespace declarations MUST be included as XML element attributes with the **rdf:RDF** root element and thus apply to the entire scope of the UAProf profile.
- Aliases for the RDF and UAProf schema namespace prefixes MUST NOT be used.
  - All profile properties (or attributes) MUST be represented as RDF property elements and MUST NOT be represented as RDF property attributes. The components in the profile are instances of the component classes identified in the UAProf schema. They MUST be identified as such by means of the **rdf:type** property of which the *resource* attribute value matches the name of the component type in the UAProf schema. For example, *MyNetworkChar* is an instantiated component of type *NetworkCharacteristics*. The profile must therefore include an RDF description such as

```
        <rdf:Description rdf:ID="MyNetworkChar">
        <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#NetworkCharacteristics"/>
```

  - Profile parsers MUST parse the profile based on the type of the component and not based on the component's name (since the name can vary, but the type is defined in the UAProf schema.) This applies to the merging of the various delta profiles ("profile-diff") that are generated or appended by different network elements.

- Each component in the profile MUST contain one or more attributes identified in the base vocabulary. It is not required for a profile to contain all the attributes in the base vocabulary. In other words, the profile instantiates a subset of the attributes in the vocabulary.

- For a given component type there MUST NOT be more than one component instance. Components MAY be fragmented within the same profile, that is the attributes of the same component instance MAY be spread across several **rdf:Description** elements. All component fragments that combine into the component instance of a particular type MUST have the same name (*ID* attribute). The **rdf:type** attribute MUST be specified with each individual fragment and MUST have the same value.

- In a fragmented component only the first fragment MAY contain a default reference **rdf:default**; all subsequent fragments, if any of the same type, MUST NOT contain a default reference.
- A defaults URL MUST refer to a UAProf profile which describes exactly one UAProf component. The component's **rdf:type** attribute MUST be specified within every default profile. Both, the referring component and the referred default component MUST have the same **rdf:type** attribute value. Defaults MUST NOT be nested, that is the referred default document itself MUST NOT contain an **rdf:default** attribute.

7.5. Profile Merging

This section describes the merging operation of two profiles A and B. Note that the merging operation is not commutative, that is the result of the merging of A onto B may differ from the result of merging B onto A.

The order of the merging is also important if more than two profiles are to be merged. In this case, the merging operation must be applied repeatedly by merging the 2nd profile onto the 1st profile, merging the 3rd profile onto the result of the previous step, and so forth until all profiles have been merged. Note that the relative order of the profiles to be merged MUST NOT be changed. The above merging process is equivalent to merging the nth profile onto the (n-1)th profile, merging the result of the previous step onto the (n-2)th profile, and so forth until all profiles have been merged.

Two profiles A and B are merged as follows:

1. A and B are normalized. A profile is in its normal form if no fragmented component instances exist and the default references are resolved in all components. If the component contains a **prf:default** element, the default property values MUST be loaded first. If the component is fragmented, the component attributes are compiled from all fragments in the order of their occurrence observing the resolution policies. These attribute values then always override eventual default values. A normalized UAProf profile therefore consists of at most six component instances without **prf:default** elements. Note that this procedure in principle would allow handling nested defaults. However, section 7.4 explicitly prohibits nesting of defaults to avoid unnecessary complexity.
2. All corresponding components are merged. Two components C_A and C_B of two profiles A and B correspond if they are instances of the same component type as indicated by the **rdf:type** attributes. Property values that are only specified in the component C_A (but not in C_B) are inserted into the result component as is. Property values that are only specified in the component of C_B (but not in C_A) are inserted into the result component as is. Property values that are specified in both components are merged according to the resolution policy associated with the particular property:

- If no resolution policy is specified or the resolution policy is **Override**, the property as specified in component C_B is inserted into the result component.
- If the resolution policy is **Locked**, the property as specified in component C_A is inserted into the result component.
- If the resolution policy is **Append**, the property value list specified in component C_B is appended to the property value list specified in component C_A, and the property with the value of the concatenated list is inserted into the result component.

3. All components of A without a corresponding component in B, are appended to the result profile as is. All components of B without a corresponding component in A, are appended to the result profile as is.

7.6. User Agent Profile Schema and Base Vocabulary

The user agent profile schema is specified in Appendix A, and tabulated in Appendix B for easy reference.

7.7. Profile Example in RDF

This section is intended to help the reader better understand the implementation and use of the specified schema and vocabulary for User Agent Profiles. The reader is expected to be familiar with RDF specifications [RDF], [RDF-Schema].

```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#">
  <rdf:Description ID="MyDeviceProfile">
    <prf:component>
      <rdf:Description ID="HardwarePlatform">
      <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#HardwarePlatform"/>
        <prf:BluetoothProfile>
          <rdf:Bag>
            <rdf:li>headset</rdf:li>
            <rdf:li>dialup</rdf:li>
            <rdf:li>lanaccess</rdf:li>
          </rdf:Bag>
        </prf:BluetoothProfile>
        <prf:ScreenSize>121x87</prf:ScreenSize>
        <prf:Model>R999</prf:Model>
        <prf:InputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>US-ASCII</rdf:li>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:InputCharSet>
        <prf:ScreenSizeChar>15x6</prf:ScreenSizeChar>
        <prf:BitsPerPixel>2</prf:BitsPerPixel>
        <prf:ColorCapable>No</prf:ColorCapable>
        <prf:TextInputCapable>Yes</prf:TextInputCapable>
        <prf:ImageCapable>Yes</prf:ImageCapable>
        <prf:Keyboard>PhoneKeypad</prf:Keyboard>
        <prf:NumberOfSoftKeys>0</prf:NumberOfSoftKeys>
        <prf:Vendor>myprofileprovider</prf:Vendor>
        <prf:OutputCharSet>
          <rdf:Bag>
            <rdf:li>ISO-8859-1</rdf:li>
            <rdf:li>US-ASCII</rdf:li>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:OutputCharSet>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
        <prf:StandardFontProportional>Yes</prf:StandardFontProportional>
      </rdf:Description>
    </prf:component>
    <prf:component>
      <rdf:Description ID="SoftwarePlatform">
      <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#SoftwarePlatform"/>
```

```
          <prf:AcceptDownloadableSoftware>No</prf:AcceptDownloadableSoftware>
        </rdf:Description>
      </prf:component>
      <prf:component>
        <rdf:Description ID="NetworkCharacteristics">
        <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#NetworkCharacteristics"/>
          <prf:SecuritySupport>
            <rdf:Bag>
              <rdf:li>WTLS-1</rdf:li>
              <rdf:li>WTLS-2</rdf:li>
              <rdf:li>WTLS-3</rdf:li>
              <rdf:li>signText</rdf:li>
            </rdf:Bag>
          </prf:SecuritySupport>
          <prf:SupportedBearers>
            <rdf:Bag>
              <rdf:li>TwoWaySMS</rdf:li>
              <rdf:li>CSD</rdf:li>
              <rdf:li>GPRS</rdf:li>
            </rdf:Bag>
          </prf:SupportedBearers>
          <prf:SupportedBluetoothVersion>1.1</prf:SupportedBluetoothVersion>
        </rdf:Description>
      </prf:component>
      <prf:component>
        <rdf:Description ID="BrowserUA">
        <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#BrowserUA"/>
          <prf:BrowserName>Ericsson</prf:BrowserName>
          <prf:CcppAccept>
            <rdf:Bag>
              <rdf:li>application/vnd.wap.wmlc</rdf:li>
              <rdf:li>application/vnd.wap.wbxml</rdf:li>
              <rdf:li>application/vnd.wap.wmlscriptc</rdf:li>
              <rdf:li>application/vnd.wap.multipart.mixed</rdf:li>
              <rdf:li>application/vnd.wap.multipart.form-data</rdf:li>
              <rdf:li>text/vnd.wap.wml</rdf:li>
              <rdf:li>text/vnd.wap.wmlscript</rdf:li>
              <rdf:li>text/x-vCard</rdf:li>
              <rdf:li>text/x-vCalendar</rdf:li>
              <rdf:li>text/x-vMel</rdf:li>
              <rdf:li>text/x-eMelody</rdf:li>
              <rdf:li>image/vnd.wap.wbmp</rdf:li>
              <rdf:li>image/gif</rdf:li>
            </rdf:Bag>
          </prf:CcppAccept>
          <prf:CcppAccept-Charset>
            <rdf:Bag>
              <rdf:li>US-ASCII</rdf:li>
              <rdf:li>ISO-8859-1</rdf:li>
              <rdf:li>UTF-8</rdf:li>
              <rdf:li>ISO-10646-UCS-2</rdf:li>
            </rdf:Bag>
          </prf:CcppAccept-Charset>
          <prf:CcppAccept-Encoding>
```

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
          <rdf:Bag>
            <rdf:li>base64</rdf:li>
          </rdf:Bag>
        </prf:CcppAccept-Encoding>
        <prf:FramesCapable>No</prf:FramesCapable>
        <prf:TablesCapable>Yes</prf:TablesCapable>
      </rdf:Description>
    </prf:component>
    <prf:component>
      <rdf:Description ID="WapCharacteristics">
      <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#WapCharacteristics"/>
        <prf:WapDeviceClass>C</prf:WapDeviceClass>
        <prf:WapVersion>2.0</prf:WapVersion>
        <prf:WmlVersion>
          <rdf:Bag>
            <rdf:li>2.0</rdf:li>
          </rdf:Bag>
        </prf:WmlVersion>
        <prf:WmlDeckSize>3000</prf:WmlDeckSize>
        <prf:WmlScriptVersion>
          <rdf:Bag>
            <rdf:li>1.2.1</rdf:li>
          </rdf:Bag>
        </prf:WmlScriptVersion>
        <prf:WmlScriptLibraries>
          <rdf:Bag>
            <rdf:li>Lang</rdf:li>
            <rdf:li>Float</rdf:li>
            <rdf:li>String</rdf:li>
            <rdf:li>URL</rdf:li>
            <rdf:li>WMLBrowser</rdf:li>
            <rdf:li>Dialogs</rdf:li>
          </rdf:Bag>
        </prf:WmlScriptLibraries>
        <prf:WtaiLibraries>
          <rdf:Bag>
            <rdf:li>WTA.Public.makeCall</rdf:li>
            <rdf:li>WTA.Public.sendDTMF</rdf:li>
            <rdf:li>WTA.Public.addPBEntry</rdf:li>
          </rdf:Bag>
        </prf:WtaiLibraries>
      </rdf:Description>
    </prf:component>
    <prf:component>
      <rdf:Description ID="PushCharacteristics">
      <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#PushCharacteristics"/>
        <prf:Push-Accept>
          <rdf:Bag>
            <rdf:li>application/wml+xml</rdf:li>
            <rdf:li>text/html</rdf:li>
          </rdf:Bag>
        </prf:Push-Accept>
        <prf:Push-Accept-Encoding>
          <rdf:Bag>
```

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
            <rdf:li>base64</rdf:li>
            <rdf:li>quoted-printable</rdf:li>
          </rdf:Bag>
        </prf:Push-Accept-Encoding>
        <prf:Push-Accept-AppID>
          <rdf:Bag>
            <rdf:li>x-wap-application:wml.ua</rdf:li>
            <rdf:li>*</rdf:li>
          </rdf:Bag>
        </prf:Push-Accept-AppID>
        <prf:Push-MsgSize>1400</prf:Push-MsgSize>
      </rdf:Description>
    </prf:component>
  </rdf:Description>
</RDF>
```

7.8. Extensions to the Schema/Vocabulary

While this specification provides a base vocabulary of property descriptions for User Agent Profiles, it is anticipated that implementers of new applications or device capabilities may, in the future, have a need for expressing new or additional attributes in the profile. The use of RDF enables an extensibility mechanism for CC/PP-based schemas that addresses the evolution of new types of devices, applications, or hardware/software. The vocabulary extensions will constitute a different RDF schema and will have a corresponding RDF model and syntactic representation. A particular profile may compose attributes from multiple schemas, namely the base vocabulary schema and the vocabulary extension schema.

The new schema/vocabulary MUST adhere to the following guidelines and recommendations for compliance with this specification and the CC/PP framework:

- New schemas and vocabularies MUST be uniquely identified using well-defined XML namespaces [XML-NS].
- Since RDF supports interoperability of schemas, the new schemas MUST NOT contain the same attributes (name and semantics) as specified in the base vocabulary. *Note that profiles can be composed from attributes from multiple schemas.* Moreover, extended attributes SHOULD NOT use attribute names that are reserved for future use by the base vocabulary, as listed in Appendix Appendix C.
- The schema vocabulary designer SHOULD have a mental model of the RDF document as an RDF graph, and should take care to express and verify the intuitions against common RDF tools. *This will eliminate potential ambiguity regarding the schema and its semantics, and interpretation by tools.*
- To reduce the verbosity of the profile and therefore the size of the HTTP headers, the CC/PP document SHOULD be expressed using the abbreviated syntax.
- Control characters and binary bytes MUST be encoded in conformance with XML syntax specifications.
- In addition to the naming conventions specified in Appendix C of the RDF Specification [RDF], the following naming conventions MUST be adhered to while defining new profile attributes and components:

  All components and attributes MUST start with an upper case letter.

  Multiple word names MUST be described as one word, with the first letter in the second word in upper case. There MUST NOT be any separator or underscore between the words. For example, Software Platform must be identified as *SoftwarePlatform*.

  Boolean values MUST be described as “Yes” and “No”.

- Cyclic references (URIs) MUST NOT be used during schema design.

7.8.1. Addition of Components

In defining components within the new schema, the designer MUST apply the following rules:

- The components used in a standardized base vocabulary MUST not only contain enough information to meet the needs of a majority of current content providers but also allow for the meaningful introduction of future device capabilities into the profile.
- As long as the new attributes fall within the realm of one of the base profile components (*HardwarePlatform*, *SoftwarePlatform*, *NetworkCharacteristics*, *WAPCharacteristics*, *PushCharacteristics*, and *BrowserUA*), the designers of new schemas must add those attributes to these defined components (instead of creating new components).
- New applications or user agents may need to assert their capabilities and preferences in a new component. The schema designer MUST uniquely define the attributes to be included in the vocabulary for the component. Attribute definition includes identifying the semantic description, attribute type (resource/ literal), the type of resolution rule and sample values.
- The schema for the new components MUST follow the general schema layout for the core profile components. Therefore, the nested description for each component MUST incorporate the following structure:

  A subordinate description block to identify default attributes if any, preferably referenced by a resource URI

  Any overrides/modifications to the default descriptions outside the Default subordinate block

7.8.2. Addition of Attributes

In defining attributes within the new schema, the designer MUST apply the following rules:

- The attributes described in the vocabulary MUST be atomic and semantically unambiguous. The names used to define/represent the attributes MUST be unique within the namespace.
- To preserve the simplicity of the schema, the use of complex data types such as containers as well as schema validation conditions such as value ranges, constraints and units SHOULD be minimized.
- A value of an attribute SHOULD be expressed as a string of characters (literal) or an RDF resource. For values that are complex data types (such as lists), RDF containers MUST be used. The **rdf:resource** construct MUST be used, where appropriate, to indicate to the RDF parser that the value of an attribute is indeed a resource and not a literal.
- The rule for resolving multiple descriptions of an attribute MUST be specified as part of the semantics. The rule must specify a *Locked*, *Override*, or *Append* treatment for value resolution. If no rule is specified, a default rule of *Override* is assumed for the attribute.

Escape control characters and binary bytes MUST be in conformance with XML syntax [XML].

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

8. Binary Encoding of User Agent Profiles

This section is normative. Examples are informative.

A User Agent Profile that is constructed in accordance with the CC/PP syntax [CCPP] and the schema of Appendix A MAY be encoded using a compact binary representation. This compact representation is based upon the WAP Binary XML (WBXML) Content Format [WBXML] and it can be used between client and WAP gateway when WSP is used, or if mandated by other specifications, such as WAP Push for OTA transmission [WAP-PushOta]. It is not expected that WBXML will be used for transmission over wireline links, or for storage of profiles, i.e. in profile repositories. This section defines how to use [WBXML] to encode a Profile document.

8.1. Token Description

8.1.1. Global Extension Tokens

This specification does not require the use of the [WBXML] global extension tokens.

8.1.2. Tag and Attribute Tokens

Section 7 defines properties for conformant User Agent Profiles, which are structured according to the CC/PP note [CCPP], and correspondingly use RDF XML serialization syntax. This section defines single-byte tokens corresponding to the elements of the RDF serialization syntax. These tokens are distributed among four code pages. Code page zero (0) defines tokens for RDF serialization elements and attributes. Code page one (1) defines tokens for properties in the core profile schema. Code page two (2) defines tokens for properties in the Browser user-agent component of the schema. Code page three (3) defines tokens for properties in the PushCharacteristics component of the schema.

User agents or applications other than the browser may define additional tag and attribute code pages for their own properties.

8.1.3. Additional Tokens

Each user agent or application that wants to define properties for use in the user agent profile MUST first define a component to hold its properties. The name of the component MUST be globally unique. Each such user agent component is considered to have a unique namespace, so that, for example, the property BackgroundColor for User Agent A is a property distinct from the property BackgroundColor for User Agent B. See Section 7 for more information.

In addition, each user agent or application SHOULD define a series of token table code pages containing the properties from its component. If it chooses to define code pages, then it SHOULD define at least two: one in the "Tag" space and one in the "Attribute" space. The property names SHOULD be inserted into each page. Any well-known values for the properties should be inserted into the "Attribute" page. Additional pages may be required if the component contains a large number of properties.

A default user agent has been defined as part of the schema: the browser. Table 8.4, Table 8.8 and Table 8.12 define the code page, two, for the browser user agent. Table 8.4 defines "Tag" code space code page two, and Table 8.8 and Table 8.12 together define "Attribute" code space code page two.

8.2. Encoding Semantics

8.2.1. XML Namespaces

WBXML does not currently support XML namespaces. As User Agent Profiles make use of namespaces, special measures must be taken to encode Profile documents using WBXML. Specifically, the XML namespaces that can be

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

used to construct Profile documents are defined and fixed by this specification. In addition, the prefixes that correspond to these available namespaces are fixed. The namespaces and their prefixes are defined in Table 8.1.

If the WBXML encoder encounters a namespace declaration that matches one of the allowed namespaces, and if its prefix does not match the defined prefix, the encoder MUST convert the document's chosen prefix to the prefix defined for that namespace in Table 8.1.

If the encoder encounters a namespace other than those defined here, it MUST encode all elements from that namespace using literal tags and MUST encode all attribute names for those elements using literal names. The namespace prefix MUST be preserved in the encoding of the tags and attribute names, and all namespace declarations (instances of the attribute xmlns) MUST be preserved.

Namespace	***URI***	***Prefix***
Resource Description Framework	http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf
Composite Capability/Preferences Profiles	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#	prf

Table 8.1: Namespace Defined for User Agent Profiles

8.2.2. Document Validation

The process of tokenizing a Profile document MUST verify that the document is well-formed according to [XML]. If it is not well-formed, encoding MUST NOT be performed. An error condition (NOT APPLIED) MAY be returned using the Profile-Warning header in this case.

8.2.3. Decoder Behavior

A decoder processing a Profile document encoded using WBXML MUST NOT consider significant the encoding means applied to a markup construct. It MUST treat a tag or attribute name encoded with a single-byte token and a tag or attribute name encoded using the LITERAL global token as equivalent if the resulting strings are equivalent.

Example: Consider the start-tag `<VoiceInputCapable>` and the following token table:

Tag Name	***Token***
`VoiceInputCapable`	`25`

Example Tag Table, Code Page 0

According to this table, the start-tag should be encoded as a single-byte tag token, 0x25.

If the above token table entry is not available to the encoder, the start-tag would be encoded using the LITERAL global token (0x04), with an argument that points to the string "VoiceInputCapable" in the string table.

A decoder processing WBXML-encoded Profile documents must treat the two encodings as equivalent.

8.3. Numeric Constants

8.3.1. Tag Tokens

8.3.1.1. RDF

The following tokens represent tags in code page zero (0). All numbers are in hexadecimal.

Tag Name	*Token*
rdf:RDF	5
rdf:Description	6
rdf:Alt	7
rdf:Bag	8
rdf:Seq	9
rdf:li	A
rdf:type	B
rdf:value	C
rdf:subject	D
rdf:predicate	E
rdf:object	F

Table 8.2: Tag Tokens, Code Page 0

8.3.1.2. Core Vocabulary

The following tokens represent tags in code page one (1). All numbers are in hexadecimal. The tag names put in parenthesis are deprecated.

Tag Name	*Token*
rdf:Description	6
rdf:Alt	7
rdf:Bag	8
rdf:Seq	9
rdf:li	A
rdf:type	B
prf:component	C
prf:defaults	D
prf:BitsPerPixel	E
prf:ColorCapable	F
prf:CPU	10
prf:ImageCapable	11
prf:InputCharSet	12
prf:Keyboard	13
prf:Model	15
prf:OutputCharSet	16
prf:PointingResolution	17
prf:ScreenSize	18
prf:ScreenSizeChar	19
prf:NumberOfSoftKeys	1A
prf:SoundOutputCapable	1B
prf:TextInputCapable	1C
prf:Vendor	1D
prf:VoiceInputCapable	1E
prf:AcceptDownloadableSoftware	1F
prf:AudioInputEncoder	20
prf:DownloadableSoftwareSupport	21
prf:JavaEnabled	22
prf:JVMVersion	23
prf:MexeClassmark	24
prf:MexeSpec	25
prf:OSName	26
prf:OSVendor	27
prf:OSVersion	28
prf:RecipientAppAgent	29
prf:SoftwareNumber	2A
prf:VideoInputEncoder	2B
prf:CurrentBearerService	2C
prf:SecuritySupport	2D

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Tag Name	*Token*
prf:SupportedBearers	2E
prf:WapDeviceClass	2F
(prf:WapPushMsgPriority)	30
(prf:WapPushMsgSize)	31
prf:WapVersion	32
prf:WmlDeckSize	33
prf:WmlScriptLibraries	34
prf:WmlScriptVersion	35
prf:WmlVersion	36
prf:WtaiLibraries	37
prf:WtaVersion	38
prf:PixelAspectRatio	39

Tag Name	*Token*
prf:StandardFontProportional	3A
(prf:WapSupportedApplications)	3B
prf:BluetoothProfile	3C
prf:MexeClassmarks	3D
prf:MexeSecureDomains	3E

Table 8.3 : Tag Tokens, Code Page 1

The following tokens represent tags in code page four (4). All numbers are in hexadecimal.

Tag Name	*Token*
prf:SupportedBluetoothVersion	10
prf:SupportedPictogramSet	11
prf:CcppAccept	12
prf:CcppAccept-Charset	13
prf:CcppAccept-Encoding	14
prf:CcppAccept-Language	15

Table 8.3.2 : Tag Tokens, Code Page 4

8.3.1.3. Browser User-Agent

The following tokens represent tags in code page two (2). All numbers are in hexadecimal. The tag names put in brackets are deprecated because they are moved to Core Vocabulary (see 8.3.1.2).

Tag Name	*Token*
rdf:Description	5
rdf:Alt	6

Tag Name	*Token*
rdf:Bag	7
rdf:Seq	8

Tag Name	*Token*
rdf:li	9
rdf:type	A
prf:component	B
prf:defaults	C
prf:BrowserName	D
prf:BrowserVersion	E
(prf:CcppAccept)	F
(prf:CcppAccept-Charset)	10
(prf:CcppAccept-Encoding)	11
(prf:CcppAccept-Language)	12

Tag Name	*Token*
prf:DownloadableBrowserApps	13
prf:FramesCapable	14
prf:HtmlVersion	15
prf:JavaAppletEnabled	16
prf:JavaScriptEnabled	17
prf:JavaScriptVersion	18
prf:PreferenceForFrames	19
prf:TablesCapable	1A
Prf:XhtmlVersion	1B
prf:XhtmlModules	1C

Table 8.4: Tag Tokens, Code Page 2

8.3.1.4. PushCharacteristics

The following tokens represent tags in code page three (3). All numbers are in hexadecimal.

Tag Name	*Token*
rdf:Description	5
rdf:Alt	6
rdf:Bag	7
rdf:Seq	8
rdf:li	9
rdf:type	A
prf:component	B
prf:defaults	C

Tag Name	*Token*
prf:Push-Accept	D
prf:Push-Accept-Charset	E
prf:Push-Accept-Encoding	F
prf:Push-Accept-Language	10
prf:Push-Accept-AppID	11
prf:Push-MsgSize	12
prf:Push-MaxPushReq	13

Table 8.5: Tag Tokens, Code Page 3

8.3.2. Attribute Start Tokens

8.3.2.1. RDF

The following tokens represent the start of an attribute in code page zero (0). All numbers are in hexadecimal.

Attribute Name	*Attribute Value Prefix*	*Token*
ID		5
rdf:about		6

Attribute Name	*Attribute Value Prefix*	*Token*
rdf:aboutEach		7
rdf:aboutEachPrefix		8

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Attribute Name	*Attribute Value Prefix*	*Token*
rdf:bagID		9
rdf:type		A
rdf:resource		B
rdf:parseType	Literal	C
rdf:parseType	Resource	D
xml:lang		E
xmlns:prf		F
xmlns:rdf		10

Table 8.6: Attribute Start Tokens, Code Page 0

8.3.2.2. Core Vocabulary

The following tokens represent the start of an attribute in code page one (1). All numbers are in hexadecimal. The attribute names put in parenthesis are deprecated.

Attribute Name	*Attribute Value Prefix*	*Token*
rdf:resource		5
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#HardwarePlatform	6
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#SoftwarePlatform	7
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#NetworkCharacteristics	8
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#WapCharacteristics	9
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#Brow	A
	serUA	
rdf:resource	http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#PushCharacteristics	B
prf:BitsPerPixel		10
prf:ColorCapable	Yes	11
prf:ColorCapable	No	12
prf:CPU		13
prf:ImageCapable	Yes	14
prf:ImageCapable	No	15
prf:InputCharSet		16
prf:Keyboard		17
prf:Model		19
prf:OutputCharSet		1A
prf:PointingResolution		1B
prf:ScreenSize		1C
prf:ScreenSizeChar		1D
prf:NumberOfSoftKeys	Yes	1E
prf:SoundOutputCapable	Yes	20

Attribute Name	***Attribute Value Prefix***	***Token***
ble		
prf:SoundOutputCapable	No	21
prf: TextInputCapable	Yes	22
prf: TextInputCapable	No	23
prf:Vendor		24
prf:VoiceInputCapable	Yes	25
prf: VoiceInputCapable	No	26
prf:PixelAspectRatio		27
prf:StandardFontProportional	Yes	28
prf:StandardFontProportional	No	29
prf:AcceptDownloadableSoftware	Yes	30
prf:AcceptDownloadableSoftware	No	31
prf:AudioInputEncoder		32
prf:DownloadableSoftwareSupport		33
prf:JavaEnabled	Yes	35
prf:JavaEnabled	No	36
prf:JVMVersion		37
prf:MexeClassmark		38
prf:MexeSpec		39
prf:OSName		3A
prf:OSVendor		3B
prf:OSVersion		3C
prf:RecipientAppAgent		3D
prf:SoftwareNumber		3E

Attribute Name	***Attribute Value Prefix***	***Token***
prf:SoundOutputCapable	No	21
prf: TextInputCapable	Yes	22
prf: TextInputCapable	No	23
prf:Vendor		24
prf:VoiceInputCapable	Yes	25
prf: VoiceInputCapable	No	26
prf:PixelAspectRatio		27
prf:StandardFontProportional	Yes	28
prf:StandardFontProportional	No	29
prf:AcceptDownloadableSoftware	Yes	30
prf:AcceptDownloadableSoftware	No	31
prf:AudioInputEncoder		32
prf:DownloadableSoftwareSupport		33
prf:JavaEnabled	Yes	35
prf:JavaEnabled	No	36
prf:JVMVersion		37
prf:MexeClassmark		38
prf:MexeSpec		39
prf:OSName		3A
prf:OSVendor		3B
prf:OSVersion		3C
prf:RecipientAppAgent		3D
prf:SoftwareNumber		3E

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Attribute Name	***Attribute Value Prefix***	***Token***
prf:VideoInputEncoder		3F
prf:CurrentBearerService		50
prf:SecuritySupport		51
prf:SupportedBearers		52
prf:WapDeviceClass		60
(prf:WapPushMsgPriority)		61
(prf:WapPushMsgSize)		62
prf:WapVersion		63
prf:WmlDeckSize		64
prf:WmlScriptLibraries		65
prf:WmlScriptVersion		66
prf:WmlVersion		67
prf:WtaiLibraries		68
prf:WtaVersion		69
(prf:WapSupportedApplications)		70
prf:BluetoothProfile		71
prf:MexeClassmarks		72
prf:MexeSecureDomains	YES	73
prf:MexeSecureDomains	NO	74
prf:SupportedBluetoothVersion		75
prf:SupportedPictogramSet		76
prf:CcppAccept		77
prf:CcppAccept-Charset		78
prf:CcppAccept-Encoding		79
prf:CcppAccept-Language		7F

Table 8.7: Attribute Start Tokens, Code Page 1

8.3.2.3. Browser User-Agent

The following tokens represent the start of an attribute in code page two (2). All numbers are in hexadecimal. The tag names put in brackets are deprecated because they are moved to Core Vocabulary (see 8.3.2.2).

Attribute Name	***Attribute Value Prefix***	***Token***
(prf:CcppAccept)		5
(prf:CcppAccept-Charset)		6
(prf:CcppAccept-Encoding)		7
(prf:CcppAccept-Language)		8
prf:DownloadableBrowserApps		9
prf:FramesCapable	Yes	A
prf:FramesCapable	No	B
prf:HtmlVersion	3.2	C
prf:HtmlVersion	4.0	D
prf:JavaAppletEnabled	Yes	E
prf:JavaAppletEnabled	No	F

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Attribute Name	***Attribute Value Prefix***	***Token***
prf:JavaScriptEnabled	Yes	10
prf:JavaScriptEnabled	No	11
prf:JavaScriptVersion		12
prf:PreferenceForFrames	Yes	13
prf:PreferenceForFrames	No	14
prf:TablesCapable	Yes	15
prf:TablesCapable	No	16
prf:XhtmlVersion		17
prf:XhtmlModules		18
prf:BrowserName		19
prf:BrowserVersion		1A

Table 8.8: Attribute Start Tokens, Code Page 2

8.3.2.4. PushCharacteristics

The following tokens represent the start of an attribute in code page three (3). All numbers are in hexadecimal.

Attribute Name	***Token***
prf:Push-Accept	5
prf:Push-Accept-Charset	6
prf:Push-Accept-Encoding	7
prf:Push-Accept-Language	8
prf:Push-Accept-AppID	9
prf:Push-MsgSize	A
prf:Push-MaxPushReq	B

Table 8.9: Attribute Start Tokens, Code Page 3

8.3.3. Attribute Value Tokens

8.3.3.1. RDF

The following tokens represent attribute values in code page zero (0). All numbers are in hexadecimal.

Attribute Value	***Token***
rdf:Statement	85
http://	86
http://www.	87
https://	88
https://www.	89
www.	8A
.com/	8B
.edu/	8C
.net/	8D
.org/	8E

Table 8.10: Attribute Value Tokens, Code Page 0

8.3.3.2. Core Vocabulary

The following tokens represent attribute values in code page one (1). All numbers are in hexadecimal.

Attribute Value	***Token***
No	85
Yes	86

Table 8.11: Attribute Value Tokens, Code Page 1

8.3.3.3. Browser User-Agent

The following tokens represent attribute values in code page two (2). All numbers are in hexadecimal

.***Attribute Value***	***Token***
No	85
Yes	86

Table 8.12: Attribute Value Tokens, Code Page 2

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

9. User Agent Profile Transport

This section is normative, unless explicitly stated otherwise. Examples are informative.

This section defines the transport mechanisms for User Agent Profile data.

From the mobile client to the WAP gateway/proxy, the User Agent profile data may be transferred over one of two protocol variants :

- Wireless Profiled HTTP; hereafter referred to as W-HTTP, or
- a combination of WSP and HTTP 1.1; hereafter referred to as WSP

A client MUST at least support one of these mechanisms in order to support UAProf. Each mechanism is functionally equivalent but a major difference is that CC/PPex is not used in the W-HTTP variant.

From the WAP gateway/proxy to the origin server the following applies: A WAP proxy (receiving W-HTTP formatted CPI information) MUST use the W-HTTP transport method towards the origin server. A WAP gateway (receiving WSP formatted CPI information) SHOULD use the W-HTTP transport method as specified in section 9.2.3.3 towards the origin server. Alternatively, CC/PPex over HTTP as specified in section 9.2.3.4 MAY be used. The use of HTTPex is not recommended. Thus, an origin server MUST be prepared to handle CPI information in either W-HTTP or CC//PPex over HTTP format.

9.1. Transport Over W-HTTP

In the case where the mobile terminal supports wireless profiled HTTP [W-HTTP] the profile is transported using meta data defined by this specification. The CC/PP Framework remains unaltered. The defined mechanism provides a functional equivalent for the CC/PP exchange protocol [CCPPex] but the definition of the syntax and semantics of the transport remains in this specification.

9.1.1. Using W-HTTP to transport CC/PP

The following extension headers are defined to transport CC/PP in W-HTTP. The defined extension headers are considered to be end to end headers. The headers are defined as general headers because they may be used in requests and responses in the push and pull use cases defined in section 6.

9.1.1.1. X-WAP-PROFILE

The *x-wap-profile* header is a general header field which MUST contain the following :

- a URI referencing the CPI or
- a reference to a profile difference, transported using the x-wap-profile-diff or
- a combination of multiple instances of these two types of data.

This data MAY be generated by the mobile terminal or attached by an intermediary point in a request to an origin server. In the case of Push this header MAY be generated as a response to a request. In the case of Push this data may be cached. However this header MUST be present in any request or response when UAProf is used. The ABNF [RFC2234] format of the header is:

Header Name:	`x-wap-profile`
Description:	List of supported references to CPI data pertaining to the terminal or intermediate performance enhancing proxies.

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Format:

```
x-wap-profile        = "x-wap-profile" ":" 1#reference

reference            = <"> ( absoluteURI | profile-diff-name ) <">

absoluteURI          = <a URI as defined by RFC2396>

profile-diff-name    = profile-diff-seq "-" profile-diff-digest

profile-diff-seq     = ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")
                        *DIGIT

profile-diff-digest  = *OCTET ";" < MD5 message digest encoded by base64 >

DIGIT                = <any US-ASCII digit "0".."9">
```

Default: None

The document referred to by absoluteURI is a profile that MAY contain several component instances, each of a different type. Components MAY contain a **prf:default** element.

The *x-wap-profile* header MAY contain references to instances of the *x-wap-profile-diff* header (defined in the following section). Each reference contains two parts, the sequence number and the profile-digest. The sequence number is used to determine the order of how the *x-wap-profile-diff* headers should be applied and the digest is used to validate that the profile-desc in the *x-wap-profile-diff* header value is correct.

The computation of the MD5 message digest takes place over the profile description defined in section 9.1.1.2, consisting of an XML/RDF document. It is introduced for the efficiency of the cache table look up in gateways, proxies and user agents and to ensure integrity of the profile description. Prior to the computation of the MD5 digest, the profile description is normalized as follows:

- Leading and trailing white spaces are eliminated. (white space as defined in RFC 2616 section 2.2: LWS)
- All non-trailing or non-leading linear white space (LWS) contained in the profile description, including line folding of multiple HTTP header lines, is replaced with one single space (SP) character.
  Note: This implies that property values, represented as XML attributes or XML element character data, MUST be adhering to white space compression as mandated in RFC 2616 section 2.2.

9.1.1.2. X-WAP-PROFILE-DIFF

The *x-wap-profile-diff* header is a general header and MAY be generated by the mobile terminal or an intermediate proxy to enhance or alter the CPI. There may be multiple profile differences, each profile difference must also have a reference in the *x-wap-profile* header which indicates the order in which differences should be applied. This header contains two parts, a sequence identifier and the entity which represents the part of the CC/PP description that is being enhanced. This header MAY be present in a request or response. In the case of Push this data may be cached.

Header Name: `x-wap-profile-diff`

Description: This header contains additional profile information which should be applied to the CPI prior to serving any content.

Format:

```
x-wap-profile-diff  = "x-wap-profile-diff" ":" profile-diff-seq

                       ";" profile-desc

profile-diff-seq    = ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")
```

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                      *DIGIT
profile-desc        =<XML document containing profile subset of
                      the schema defined in Section 7>
DIGIT               = <any US-ASCII digit "0".."9">
```

Default: None

The document specified in profile-desc is a profile that MAY contain several component instances, each of a different type. Components MAY contain a **prf:default** element.

The actual profile referred to in the x-wap-profile header URI is merged with its correlated x-wap-profile-diff header values in the order in which they occur. This process is applied to all URIs contained in the x-wap-profile header. Finally, the resulting profiles are merged in the order in which the corresponding URIs were defined in the x-wap-profile header.

The XML/RDF document which forms the "x-wap-profile-diff" header value MUST be adhering to linear white space (LWS) substitution rules as per section 2.2 of RFC 2616.

9.1.1.3. X-WAP-PROFILE-WARNING

The x-wap-profile-warning header is a general header. Its presence indicates the level to which the response has been tailored in relation to profile data that has been supplied in the request. This header MAY be present in a request or response.

Header Name: `x-wap-profile-warning`

Description: This header is used by the server to indicate whether the CPI has been honoured when the response to the request was generated.

Format:

```
x-wap-profile-warning = "x-wap-profile-warning" ":" warning-code
warning-code        =  200 | 201 | 202 | 203 | 500
```

Default: None

The warning codes that are defined fall into the following categories :

- 1xx – reserved
- 100 -- reserved
- 2xx – indicates whether the content has been adapted depending on the profile
- 5xx – indicates the server is incapable of processing CPI.

The *x-wap-profile-warning* may have the following values.

200 Not applied

This value MUST be included if the content has not been tailored, and is sent in a representation which is the only representation available in the server.

201 Content selection applied

MUST be included if the included content has been selected from one of the representations available.

202 Content generation applied

© 2001, Wireless Application Protocol Forum, Ltd. All rights reserved

MUST be included if the content has been tailored or generated as a result of applying the included profile.

203 Transformation applied

MUST be added by an intermediate proxy if it applies any transformation changing the content-coding based on the CPI data.

500 Not Supported

Indicates that the entity sending this warning code does not support UAProf.

9.1.2. Protocol Procedures

9.1.2.1. Over The Air

In this transport variant the headers and their values are not compressed for over the air transmission. It is recommended that the hardware and software manufacturer only use the *x-wap-profile* header to indicate an `absoluteURI` as the reference for CPI for the mobile terminal when transmitted over the air. However this specification does not preclude the use *x-wap-profile-diff* in this case.

9.1.2.2. Combining X-WAP-Profile and X-WAP-Profile-Diff

If the *x-wap-profile-diff* header is included the *profile-diff-seq* MUST match a sequence number in the *x-wap-profile* header otherwise the associated profile-diff-desc MUST NOT be processed. The reference to each *x-wap-profile-diff* header contains two parts, a sequence number which governs the order in which the *x-wap-profile-diff* headers are processed and an MD5 message digest which is used to validate the profile-desc which is contained in the *x-wap-profile-diff.* If either the sequence or the MD5 validation do not match, the particular profile-diff MUST be ignored.

If the *x-wap-profile-diff* header is added by an intermediate proxy, it MUST NOT alter the existing sequence of *x-wap-profile-diff* headers, the proxy MUST append using the next available sequence number in numeric order.

The digest associated with an *x-wap-profile-diff* MUST be generated by applying MD5 message digest algorithm [RFC1321] and Base64 algorithm, section 6.8 in the MIME specification [RFC2045] to the corresponding *profile-desc* part of the header field-value. The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. The Base64 algorithm takes as input arbitrary binary data and produces as output printable encoding data of the input.

9.1.2.3. Relationship with HTTP Headers

The profile information referred to in the *x-wap-profile* and *x-wap-profile-diff* header does not supersede HTTP request or response header information.

9.1.2.4. Caching

The *x-wap-profile-warning* header MUST NOT be used for cache control purposes. If a server wishes to indicate a caching dependency based on these headers then it should use the Vary header as defined in section 14.44 of the HTTP 1.1 specification [HTTP].

9.1.2.5. Examples

This section is informative

Use Case : HTTP Request	
Mobile Terminal	*Performance Enhancing Proxy*
Request → Origin Server	
GET http://anyuri/ HTTP/1.1 Host: plaintext x-wap-profile: "http://profilerepository/" …	GET http://anyuri/ HTTP/1.1 Host: plaintext x-wap-profile: "http://profilerepository/", “1-uKdjJHuhjHUuj” x-wap-profile -diff:1; <?xml…./> …
Mobile Terminal	*Origin Server*
	Response ←
	HTTP/1.1 200 OK Date: x-wap-profile -warning:202 …

Use Case : Push Proxy Gateway Request for Profile	
Mobile Terminal	*Push Proxy Gateway*
	← Request
	OPTIONS * HTTP/1.1 Host: …
Response →	
HTTP/1.1 204 OK Date: x-wap-profile: "http://profilerepository/"	

Use Case : Push of Tailored Content

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Mobile Terminal	*Push Proxy Gateway*
	← Request
	POST /wappush HTTP/1.1 Host: x-wap-profile -warning:200 …

9.2. User Agent Profile Transport Over WSP

This section specifies how Profiles are transported over WSP. Section 9.2.1 is an introduction and is not normative. The CC/PP Exchange Protocol over WSP, referred to as CC/PP-WSP, is specified in the normative Sections 9.2.2 and 9.2.3.

9.2.1. Introduction

9.2.1.1. The CC/PP Framework and the CC/PP Exchange Protocol Over HTTP

The Composite Capability/Preference Profiles (CC/PP) defines a framework for content negotiation [CCPP]. Section 7 defines a vocabulary of categories and attributes for WAP-enabled devices, and Section 8 describes how the CC/PP document is encoded using the WAP Binary XML [WBXML] standard.

To transport CC/PP documents, or references to such profiles, over the Internet, the CC/PP Exchange Protocol Over HTTP MAY be used. It is specified in [CCPPEx]. The CC/PP Exchange Protocol Over HTTP, referred to as CC/PP-HTTP, is used over HTTP [HTTP] and uses the HTTP Extension Framework [HTTPext]. The mapping onto WSP is specified in this section and is sometimes referred to as CC/PP-WSP.

The CC/PP Exchange Protocol specifies two new request header fields (*Profile*, *Profile-Diff*) and one new response header field (*Profile-Warning)*. The *Profile* header is used to transport one or many Profile identities, URIs, from the client to the server. This set of Profiles is used to construct the Composite Profile Information (CPI). The *Profile-diff* header is used to transport changes to the CPI. This means that the *Profile-Diff* header must always be used together with and referenced by the *Profile* header. The *Profile-Warning* header is used by the server to notify the client whether the request to use Profiles was fulfilled, partly fulfilled, or not fulfilled. To extend the HTTP protocol with the new header in a structured way, the HTTP Extension Framework MAY be used [HTTPext], however the headers defined in Section 9.1 SHOULD be used to transport CPI.

9.2.1.2. Using WSP to Transport CC/PP Profiles

The WSP protocol has some features that cannot be found in HTTP. To reduce the size of request messages the WSP client can cache headers in the gateway for the lifetime of a WSP session. The cached headers are called session headers and are sent to the gateway during session establishment. The client can use the **Resume** operation of WSP to update the session headers during the session. At any time the client or the gateway can terminate the session and establish a new one, with new session headers. Moreover, the client MAY provide additional headers with each request; these request headers are merged with the cached headers (and, possibly, other WSP headers) to generate the final CPI that is transmitted over HTTP.

WSP uses *Profile* and *Profile-Diff* headers to convey the CPI. A *Profile* header contains a single URL, referencing an externally accessible CPI document. The *Profile-Diff* header contains a WBXML-encoded CPI document. Multiple *Profile* and *Profile-Diff* headers MAY be cached by the gateway and/or included with a request.

The WAP gateway combines request headers with cached session headers to create HTTP requests [WAE]. The following list summarizes the WSP header management:

- If one or many *Profile* headers are cached in the server, the client can override all of them within the scope of a particular request by sending one or more *Profile* headers in a single request message

- If one or many *Profile* headers, but no *Profile-Diff* headers, are cached in the gateway, the client can append one or many *Profile-Diff* headers within the scope of a particular request by including them in a request message.
- If one or many Profile-Diff headers are cached in the gateway, the client can override all of them within the scope of a particular request by sending one or more Profile -Diff headers in a single request message.
- If one or many *Profile-Diff* headers, but no *Profile* headers, are cached in the gateway, the client can append one or many *Profile* headers within the scope of a particular request by including them in a request message.
- By using the Resume function, the client can update the session headers, without establishing a new session[3].

9.2.1.3. Differences Between CC/PP-HTTP and CC/PP-WSP

The following are the differences between CC/PP-HTTP and CC/PP-WSP:

- In the CC/PP-WSP *Profile-Warning* response header, the warning text is not included.
- In CC/PP-HTTP, multiple profile references are transmitted in one *Profile* header. The header may reference both external profiles (via a URL) or embedded profiles (via a URN containing an MD5 checksum of the embedded profile). Embedded profiles are associated with custom headers computed from the profile's MD5 checksum. In WSP, a *Profile* header can only transmit one profile reference, but multiple *Profile* headers can be transmitted in the same WSP header; the *Profile* header only references external profiles via URL. In addition, multiple *Profile-Diff* headers, each containing an embedded profile, may be transmitted in the same WSP header. No functionality is lost, because the WAP gateway is capable of generating the MD5 checksums for the Profile-Diff documents and constructing a complete *Profile* header for transmission over HTTP. In the CC/PP-WSP *Profile-Diff* header, the profile section must be encoded using WBXML as specified in Section 8. In the CC/PP-HTTP *Profile-Diff* header, the profile section must be sent as XML text.
- In the CC/PP-WSP *Profile-Diff* header, the profile section must be encoded using WBXML as specified in Section 8. In the CC/PP-HTTP *Profile-Diff* header, the profile section must be sent as XML text.

9.2.2. Structure and Encoding of Header Fields

9.2.2.1. The Profile Header

The syntax of the Profile header SHOULD conform to the production of [**1.**].

[1.] Profile = Profile-field-name Profile-field-value

[2.] Profile-field-name = Short-integer [WSP]

[3.] Profile-field-value = Uri-value

Example:

0xb5 http://anyco.com/anypda 0x00

In the above example the profile URI is http://profile.anyuri.com/anypda. Note that 0x00 signals the end of the URI-value string.

[3] This is not possible in WSP 1.0.

© 2001, Wireless Application Protocol Forum, Ltd. All rights reserved

9.2.2.2. The Profile-Diff Header

The syntax of the Profile-Diff header MUST conform to the production of [**4.**]. The CC/PP-profile in production [**7.**] MUST be encoded using WBXML as specified in Section 8.

[4.] Profile-diff = Profile-diff-field-name Profile-diff-field-value

[5.] Profile-diff-field-name = Short-integer [WSP]

[6.] Profile-diff-field-value = Value-length CCPP-profile

[7.] CCPP-profile = *Octet; encoded using WBXML [WBXML]

<u>Example:</u>

0xb6 0x0A 0x01 0x05 0x04 ...

In the above example:

- The length is 10 octets: 0x0A;
- The binary XML version is 1.1: 0x01;
- The public identifier has been assigned the value of 5: 0x05; and
- The character set is iso-8859-1 : 0x04

9.2.2.3. The Profile-Warning Header

The syntax of the Profile-Warning header MUST conform to the production of [**8.**].

[8.] Profile-warning = Profile-warning-field-name Profile-warning-value

[9.] Profile-warning-field-name = Short-integer [WSP]

[10.] Profile-warning-value = Warn-code | (Value-length Warn-code Warn-target *Warn-date)

[11.] Warn-code = Short-integer

Status codes (and corresponding Short-integer values) are 100 (0x90) | 101 (0x91) | 102 (0x92) | 200 (0xa0) | 201 (0xa1) | 202 (0xa2) | 203 (0xa3) [CCPPEx]

[12.] Warn-target = Uri-value | host [“:” port]

[13.] Warn-date = Date-value

<u>Example 1:</u>

0xb7 0x16 0x92 http://anyco.com/pda 0x00

In the above example:

- The length is 22 octets (0x16)
- The profile warning code is 102 (encoded as 0x92); and
- The profile URI is http://profile.anyuri.com/anypda
- No date is provided

Example 2:

```
0xb7 0x92
```

In the above example, only the warning code is sent.

9.2.3. Protocol Procedures

9.2.3.1. Session Establishment

The following procedure is used to establish a WSP session that uses User Agent Profiles:

[14.] To indicate support for User Agent Profiles, the client MUST include the *Profile* header in the connect message.

[15.] **If the server responds with the Profile Warning Header (set to warning code 100), the gateway MUST forward this to the client.**

[16.] If the client does not receive the *Profile-Warning* header with the warning code 100, it MUST NOT send any additional User Agent Profile headers during the WSP session.

During the session the client can use the Resume procedure to update the session header [WAE] [4].

Session headers are cached in the gateway for the duration of the WSP session [WSP]. When the gateway generates HTTP request messages it combines the headers from the request message with the headers in the session's cache, according to the rules in section 9.2.3.2.

9.2.3.2. Combining Session and Request Headers

Upon reception of a WSP request, the WAP gateway combines the request headers with the cached WSP session headers. The result is a list of WSP headers that gets translated into one HTTP request, see section 9.2.3.3. The following procedure is used to combine the request headers with the cached session headers into one list of headers:

[4] This is not possible in WSP version 1.0.

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

[17.] Headers from the WSP request MUST override cached WSP session headers according to the rules for client headers defined in [WAE].

[18.] If, after the previous operation, both request headers and session headers are present in the list of headers, request headers MUST follow after session.

Since request headers are appended to the list after the session headers, request headers will take precedence over session headers if the headers are applied to the profile in the same order as they are transported.

Example 1:

In this example, both request and session headers remain after the WSP request and the WSP session have been combined. In the result, the request header comes after the session headers in the list.

CC/PP cached session headers:

```
Profile: URIx
Profile: URIy
```

CC/PP request header:

```
Profile-Diff: 0x01 0x05 0x04 0xBB
```

Will result in:

```
Profile: URIx
Profile: URIy
Profile-Diff: 0x01 0x05 0x04 0xBB
```

Example 2:

In this example, the request headers override all session headers.

CC/PP cached session headers:

```
Profile: URIx
Profile: URIy
```

CC/PP request header:

```
Profile: URIz
Profile-Diff: 0x01 0x05 0x04 0xBB
```

Will result in:

```
Profile: URIz
Profile-Diff: 0x01 0x05 0x04 0xBB
```

9.2.3.3. Header Translation Between CC/PP-WSP and HTTP

Optionally, the WAP gateway MAY forward WSP requests as HTTP 1.1 requests [WAE]. In forwarding the request, the gateway MUST forward all CC/PP-WSP headers (defined in Section 9.2.2 and resolved at the gateway according to the rules of Section 9.2.3.2) as W-HTTP headers (defined in section 9.1) according to these rules:

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

[19.] Each CC/PP-WSP Profile-Diff header field is translated into exactly one W-HTTP *x-wap-profile-diff* header field. The HTTP Profile-Diff headers are generated dynamically as specified in [CCPPex]. WBXML encoding of the profile section MUST be decoded, leaving the profile section as XML text.

[20.] A single W-HTTP x-wap-profile header field is generated as specified in [CCPPex] by listing the values of each CC/PP-WSP *Profile* header and the values of each dynamically generated W-HTTP *x-wap-profile-diff* header. The ordering of this list preserves the ordering of the corresponding WSP *Profile* and/or *Profile-Diff* headers in the WSP setup/Resume and WSP request messages, as appropriate.

[21.] One W-HTTP *x-wap-profile-warning* response header is translated into exactly one CC/PP-WSP *Profile-Warning* response header. The warning text from the W-HTTP header is not translated.

[22.] The client that wants to convey Accept header information MUST do so through standard WSP headers, such as Accept, Accept-Charset, and Accept-Language. Information contained in these headers MUST constitute part of the CPI.

In forwarding the HTTP request and generating the W-HTTP x-wap-profile and x-wap-profile-diff headers, the gateway MAY insert additional profile information into the request. If provided, this additional information MUST be presented by appending to the end of the *x-wap-profile* header either a URI or a dynamically generated *x-wap-profile-diff* header identifier. Accordingly, a compliant gateway MAY therefore introduce an *x-wap-profile* (and, if necessary, *x-wap-profile-diff* header) on behalf of a client whose WSP session has no cached *Profile* or *Profile-Diff* headers. This support enables a gateway to support User Agent Profiles on behalf of client devices that are otherwise unable to convey profile information.

Example:

The WSP headers:

```
Profile: URIx
Profile-Diff: 0x01 0x05 0x04 0xAA
Profile: URIy
Profile-Diff: 0x01 0x05 0x04 0xBB
```

Are translated into the following HTTP headers:[5]

```
x-wap-profile: “URIx”, “1-uKdjJHuhjHUuj”, “URIy”, “2-jdsjhUHjsuHjU”
x-wap-profile-diff: 1;<?xml version="1.0"?><RDF>AA...
x-wap-profile-diff: 2;<?xml version="1.0"?><RDF>BB...
```

[5] The value of the *Profile-diff* digest is not real, see [CCPPex].

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

10. Origin Server Behaviour

This section is informative.

From the gateway to the origin server, the User Agent Profile is transported over the Internet. In the WAP architecture specification, and as specified in Section 9.2.3.3, HTTP is assumed to be the transport mechanism for Internet-related information, using the HTTP 1.1 mechanisms for header management and caching [WAE].

The transmission of profiles information is described in section 9.

A combination of the following components may be necessary to implement an Internet server capable of receiving User Agent Profile information:

- A HTTP 1.1 server [HTTP]
- The HTTP 1.1 Extension Framework [HTTPext]
- The CC/PP Exchange Protocol [CCPPex]

Upon receiving a User Agent Profile, an origin server may do the following (architecture is described in Figure 5.1):

- Retrieve profile from the profile repository
- Parse the profile
- Validate the syntax of the profile
- If error condition occurs inform gateway/proxy with the proper error code as specified in Section 0
- Resolve attribute values by applying overriding rules and default values, applying the algorithm described in Section 6.4
- Validate the attribute value types

Customize content according to the information contained within the profile

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

11. Deployment Considerations

This section is informative.

End-to-end systems supporting User Agent Profiles will depend upon the support of many of the elements in the WAP architecture:

- Client devices may need to store, generate, and transmit profile information.
- Gateways need to forward profile information to proxies or servers.
- Proxies may need to modify some of the profile information according to services that they provide, or they may need to tailor their provided services according to the client's profile information. They may also provide profile persistence or caching services.
- Servers may need to utilize the profile information to help adapt content that is to be provided to the client device.

A variety of schemes may be used to provision and maintain the profile information. Designation of these methods and approaches is beyond the scope of this specification. However, this section outlines some concepts that may need to be taken into consideration in the preparation and deployment of a UAProf capable system.

The possible uses of CC/PP are described by the W3C in [CCPP].

11.1. Client Support

It is clear that this specification will not be applied to initial WAP products which conform to the WAP 1.1 specifications. For backward compatibility, therefore, future systems will need to support clients and gateways that are unaware of User Agent Profiles. Similarly, support for one-way and broadcast service paradigms will create service models different than that for web browsing. Generally, therefore, client device support for User Agent Profiles will take various forms and will need to be supported in a variety of ways.

11.1.1. Client Devices Not Supporting User Agent Profiles

For those devices that do not directly support or cannot transmit UAProf information, indirect support may be provided by the gateway. It may be possible to have a static profile provisioned at the gateway by a carrier or service provider. This profile would be presented by WAP gateways to proxies and servers on behalf of the device(s) involved. The client device in this case does not require any special provisioning or support for this service.

The WAP gateway may also support dynamic, customized profiles on behalf of these legacy devices and one-way devices. For example, the WAP gateway may apply a dynamically generated profile according to the particular device ID or subscriber invoking the service.

11.1.2. Client Devices Supporting User Agent Profiles

For those devices that provide UAProf capabilities, the support may take different forms and therefore demand various provisioning schemes. Some devices may only support a static profile header which is transmitted during WSP session invocation or on sending HTTP request. Other devices may also be able to collect and send user preference information during the WSP session invocation or on sending HTTP request.

11.1.2.1. Static Header Support

For some client devices, a fixed header may be utilized to support the UAProf *Profile* header reported during WSP session creation or on sending HTTP request by using the HTTP extension framework. And for some other client devices, a fixed header may be utilized to support the UAProf *x-wap-profile* header reported on sending HTTP request. This fixed header may be common for all devices of the same model or may be somewhat individualized for each device. A profile could be loaded into non-volatile memory by the manufacturer, network operator, or service provider. Typically, this profile would simply be a URI reference to a shared set of preference data stored on some repository on

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

behalf of all such devices. This shared set of profile information would likely be deployed by the manufacturer, network operator, or service provider to cover the service characteristics that they wish to enable.

Depending on the number of devices that share the static profile, the supplied URI may itself become a de-facto preference indicator for proxies or origin servers. For example, a reference to a profile at http://profile.anyuri.com/anypda that would be invoked by millions of client devices eventually could lead to content shaping operations based upon the URI itself rather than upon the particular profile attributes that the URI references. This could lead to a misuse of the UAProf capability because, strictly speaking, the profile referenced by the URI could change arbitrarily at its server, thereby leading to incorrect behavior at the proxies or origin servers. Consequently, once a URI emerges as a moniker to represent a default profile, care must be taken not to modify the profile that the URI references.

An individualized URI for the static profile could be stored on each device. This would provide the opportunity to have some particular device-specific information stored within the profile.

Alternatively, the entire static profile may be stored on the client. The definition of the values in such a schema may be performed in several ways: preset by the manufacturer during production; hardcoded in the user agent software; programmed by the network operator or service provider at time of subscriber lease/purchase; and/or, programmed over-the-air by the network operator or service provider. The methods that may be developed are beyond the scope of this specification.

11.1.2.2. Session-Constant Support

The client device may have the capability to dynamically construct its profile and present that profile during the WSP session creation. The profile would be used throughout the WSP session lifetime.

The profile attributes delivered by the device may be user alterable or may be pre-set by the device manufacturer, network operator, or service provider. For example, a device may permit the user to select the default language (e.g. English, French, or German) but may prevent the user from altering certain other profile attributes, including the URI representing the basic characteristics of the device or network. The techniques for changing these attributes may differ: a user interface or form may be used for the user-modifiable values, and an over-the-air scheme may be used for the server-managed values. The methods that may be developed are beyond the scope of this specification.

11.1.2.3. Request-Specific Support

For more dynamic profile updates during the WSP session, the client device needs to support the *Profile-Diff* header scheme. And when using UAProf over W-HTTP, each profile is Request-Specific.

For example, a user agent, like a browser, may allow the subscriber to indicate a preference for images. By changing the preference, the user agent would have to send an update over the WSP session indicating the change so that it could be included in the cached profile at the WAP gateway. Depending on the user interface model of the user agent, the change may be temporary (for the current WSP request) or more long term (service toggle). These would be reflected in whether the new attribute value is sent within a *Profile-Diff* included in a WSP request or a WSP **Resume** operation.

The device may update the cached session profile without direct user intervention. For example, a phone could be designed to block a user agent's access to the audio services while a call is in progress. When a telephone call is made, the device may send an updated Profile-Diff to the WAP gateway or the Origin Server to indicate that the user agent no longer has access to audio capabilities.

11.2. Repository Support

A profile repository typically would be an HTTP server that provides UAProf CPI elements upon request. UAProf profiles may reference data stored in repositories provided and operated by the subscriber; network operator; gateway operator, device manufacturer; or service provider. It is expected that there will be a variety of mechanisms put into place to create and manage the data in such repositories:

- Manufacturers and software vendors may provide static profile resources that describe the client devices that they produce. Such profiles will describe the hardware and standard software elements that exist on the client devices.
- Network operators may provide profile information that includes network characteristics, and gateway operators may provide profile information that defines permitted service levels through the communications infrastructure. Some of these operators may provide additional services enabling the storage of user preference information.
- Subscribers may look to other entities to store preference information.
- Service providers, including enterprise IT managers, may create profile repositories that reflect the needs of their hosted applications or services.

Independent of the content of the stored profile, policy controls may be imposed that limit what profile information is delivered to a particular requester. Any such limits, or methods employed to ensure compliance, are beyond the scope of this specification.

The specification of an architectural element that can be used as a profile repository is not in the scope of this document.

11.3. Interim Proxy Support

The CC/PP working group describes in [CCPP] how proxies can be used to do content adaptation and transformation.

Appendix A. User Agent Profile Schema

This section is normative.

This section specifies the schema and base vocabulary for WAP User Agent Profiles. The schema is expressed in RDF and encoded in XML, and it includes semantic descriptions of all attributes identified in the vocabulary. However, a profile that conforms to the schema is not required to instantiate all attributes contained in the schema. Because of limitations in the existing XML and RDF standards, certain information in the schema such as attribute type is currently described in the form of RDF or XML comments. For the purposes of determining whether a profile conforms to this specification, the entire schema – including comments – is considered normative. *In future versions of this specification, information currently contained within comments is likely to be described more formally using standard XML and/or RDF methods.*

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
                  xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#">

                <rdf:Description ID="Component">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
                                <rdfs:label>Component</rdfs:label>
                                <rdfs:comment>
      A Component within the CC/PP Schema is a class of related properties
      that describe the capabilities and preferences information.
   </rdfs:comment>

                </rdf:Description>
                <!-- ****************************************************************** -->
                <!-- ***** Properties shared among the components***** -->

                <rdf:Description ID="component">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:label>component</rdfs:label>
                                <rdfs:comment>
      The component attribute links the various components to the root node
      (profile).
    </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="defaults">
                                <rdfs:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
                                <rdfs:domain rdf:resource="#PushCharacteristics"/>
                                <rdfs:comment>
      An attribute used to identify the default capabilities.
   </rdfs:comment>
                </rdf:Description>

                <!-- ****************************************************************** -->
                <!-- ***** Component Definitions ***** -->

                <rdf:Description ID="HardwarePlatform">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: HardwarePlatform</rdfs:label>
                                <rdfs:comment>
      The HardwarePlatform component contains properties of the device's
      Hardware, such as display size, supported character sets, etc.
   </rdfs:comment>
                </rdf:Description>
```

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                <rdf:Description ID="SoftwarePlatform">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: SoftwarePlatform</rdfs:label>
                                <rdfs:comment>
      The SoftwarePlatform component contains properties of the device's
      application environment, operating system, and installed software.
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="BrowserUA">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: BrowserUA</rdfs:label>
                                <rdfs:comment>
      The BrowserUA component contains attributes related to the browser
      user agent running on the device.
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="NetworkCharacteristics">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: NetworkCharacteristics</rdfs:label>
                                <rdfs:comment>
      The NetworkCharacteristics component contains properties describing the
      network environment including the supported bearers.
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WapCharacteristics">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: WapCharacteristics</rdfs:label>
                                <rdfs:comment>
      The WapCharacteristics component contains properties of the WAP
      environment supported by the device.
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="PushCharacteristics">
                                <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
                                <rdfs:subClassOf rdf:resource="#Component"/>
                                <rdfs:label>Component: PushCharacteristics</rdfs:label>
                                <rdfs:comment>
      The PushCharacteristics component contains properties of the device's
      push capabilities, such as supported content mime types.
   </rdfs:comment>
                </rdf:Description>

                <!-- **
     ** In the following property definitions, the defined types
     ** are as follows:
     **
     **    Number:     A positive integer
     **                  [0-9]+
     **    Boolean:    A yes or no value
     **                  Yes|No
     **    Literal:    An alphanumeric string
     **                  [A-Za-z0-9/.\-_]+
     **    Dimension:  A pair of numbers
     **                  [0-9]+x[0-9]+
     **
-->
                <!-- ****************************************************************** -->
                <!-- ***** Component: HardwarePlatform ***** -->

                <rdf:Description ID="BluetoothProfile">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
                Description: Supported Bluetooth profiles as defined in the Bluetooth specification
[BLT].
                Type:                            Literal (bag)
                Resolution:      Locked
                Examples:        "dialup", "lanAccess"
          </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="BitsPerPixel">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  The number of bits of color or grayscale information per
                    pixel, related to the number of colors or shades of gray
                    the device can display.
      Type:         Number
      Resolution:   Override
      Examples:     "2", "8"
</rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="ColorCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device's display supports color.
                    "Yes" means color is supported. "No" means the display
                    supports only grayscale or black and white.
      Type:         Boolean
      Resolution:   Override
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="CPU">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Name and model number of the device CPU.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Pentium III", "PowerPC 750"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="ImageCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device supports the display of
                    images. If the value is "Yes", the property CcppAccept
                    may list the types of images supported.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="InputCharSet">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  List of character sets supported by the device for text
                    entry. Property's value is a list of character sets,
                    where each item in the list is a character set name, as
                    registered with IANA.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "US-ASCII", "ISO-8859-1", "Shift_JIS"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="Keyboard">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Type of keyboard supported by the device, as an indicator
                    of ease of text entry.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Disambiguating", "Qwerty", "PhoneKeypad"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="Model">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Model number assigned to the terminal device by the
                    vendor or manufacturer.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Mustang GT", "Q30"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="NumberOfSoftKeys">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Number of soft keys available on the device.
      Type:         Number
      Resolution:   Locked
      Examples:     "3", "2"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="OutputCharSet">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  List of character sets supported by the device for
                    output to the display. Property value is a list of
                    character sets, where each item in the list is a
                    character set name, as registered with IANA.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "US-ASCII", "ISO-8859-1", "Shift_JIS"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="PixelAspectRatio">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Ratio of pixel width to pixel height.
      Type:         Dimension
      Resolution:   Locked
      Examples:     "1x2"
   </rdfs:comment>
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                </rdf:Description>

                <rdf:Description ID="PointingResolution">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Type of resolution of the pointing accessory supported
                    by the device.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Character", "Line", "Pixel"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="ScreenSize">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  The size of the device's screen in units of pixels,
                    composed of the screen width and the screen height.
      Type:         Dimension
      Resolution:   Locked
      Examples:     "160x160", "640x480"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="ScreenSizeChar">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Size of the device's screen in units of characters,
                    composed of the screen width and screen height. The
                    device's standard font should be used to determine
                    this property's value. (Number of characters per
                    row)x(Number of rows). In calculating this attribute
                    use the largest character in the device's default font.
      Type:         Dimension
      Resolution:   Locked
      Examples:     "12x4", "16x8"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="StandardFontProportional">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device's standard font is
                    proportional.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="SoundOutputCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform "/>
                                <rdfs:comment>
      Description:  Indicates whether the device supports sound output
                    through an external speaker, headphone jack, or other
                    sound output mechanism.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>
```

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                <rdf:Description ID="TextInputCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device supports alpha-numeric text
                    entry. "Yes" means the device supports entry of both
                    letters and digits. "No" means the device supports only
                    entry of digits.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="Vendor">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Name of the vendor manufacturing the terminal device.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Ford", "Lexus"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="VoiceInputCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#HardwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device supports any form of voice
                    input, including speech recognition. This includes voice-
                    enabled browsers.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>
                <!-- ****************************************************************** -->
                <!-- ***** Component: SoftwarePlatform ***** -->

                <rdf:Description ID="AcceptDownloadableSoftware">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates the user's preference on whether to accept
                    downloadable software.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="AudioInputEncoder">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of audio input encoders supported by the device.
      Type:         Literal (bag)
      Resolution:   Append
      Example:      "G.711"
   </rdfs:comment>
                </rdf:Description>

<rdf:Description ID="CcppAccept">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of content types the device supports. Property
                    value is a list of MIME types, where each item in the
                    list is a content type descriptor as specified by
                    RFC 2045.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "text/html", "text/plain", "text/html", "image/gif"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="CcppAccept-Charset">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of character sets the device supports. Property
                    value is a list of character sets, where each item in
                    the list is a character set name registered with IANA.

      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "US-ASCII", "ISO-8859-1", "Shift_JIS"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="CcppAccept-Encoding">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of transfer encodings the device supports.
                    Property value is a list of transfer encodings, where
                    each item in the list is a transfer encoding name as
                    specified by RFC 2045 and registered with IANA.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "base64", "quoted-printable"
  </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="CcppAccept-Language">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Seq"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of preferred document languages. If a resource is
                    available in more than one natural language, the server
                    can use this property to determine which version of the
                    resource to send to the device. The first item in the
                    list should be considered the user's first choice, the
                    second the second choice, and so on. Property value is
                    a list of natural languages, where each item in the list
                    is the name of a language as defined by RFC 3066[RFC3066].
      Type:         Literal (sequence)
      Resolution:   Append
      Examples:     "zh-CN", "en", "fr"
   </rdfs:comment>
                </rdf:Description>


                <rdf:Description ID="DownloadableSoftwareSupport">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
```

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
      Description:  List of executable content types which the device
                    supports and which it is willing to accept from the
                    network. The property value is a list of MIME types,
                    where each item in the list is a content type
                    descriptor as specified by RFC 2045.
      Type:         Literal (bag)
      Resolution:   Locked
      Examples:     "application/x-msdos-exe"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JavaEnabled">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Indicates whether the device supports a Java virtual
                    machine.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JavaPlatform">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description: The list of JAVA platforms and profiles installed in the device. Each item in the
list is a name token describing compatibility with the name and version of the java platform
specification or the name and version of the profile specification name (if profile is included in
the device)

      Type:         Literal (bag)
      Resolution:   Append
      Examples: "Pjava/1.1.3-compatible", "MIDP/1.0-compatible", "J2SE/1.0-compatible"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JVMVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of the Java virtual machines installed on the
                    device. Each item in the list is a name token describing
                    the vendor and version of the VM.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "SunJRE/1.2", "MSJVM/1.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="MexeClassmarks">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform "/>
                                <rdfs:comment>
Description: List of MExE classmarks supported by the device. Value "1" means the MExE device
supports WAP, value "2" means that MExE device supports Personal Java and value "3" means that MExE
device supports MIDP applications.
                Type:         Literal (bag)
                Resolution:   Locked
                Examples:     "1", "3"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="MexeSpec">
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Class mark specialization. Refers to the first two
                    digits of the version of the MExE Stage 2 spec.
      Type:         Literal
      Resolution:   Locked
      Examples:     "7.02"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="MexeSecureDomains">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
                 Description: Indicates whether the device's supports MExE security domains. "Yes"
means that security domains are supported in accordance with MExE specifications identified by the
MexeSpec attribute. "No" means that security domains are not supported and the device has only
untrusted domain (area).
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="OSName">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Name of the device's operating system.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Mac OS", "Windows NT"
   </rdfs:comment>
                </rdf:Description>
                <rdf:Description ID="OSVendor">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Vendor of the device's operating system.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Apple", "Microsoft"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="OSVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Version of the device's operating system.
      Type:         Literal
      Resolution:   Locked
      Examples:     "6.0", "4.5"
   </rdfs:comment>
                </rdf:Description>
                <rdf:Description ID="RecipientAppAgent">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  User agent associated with the current request. Value
                    should match the name of one of the components in the
                    profile. A component name is specified by the ID
                    attribute on the prf:Component element containing the
                    properties of that component.
      Type:         Literal
      Resolution:   Locked
      Examples:     "BrowserMail"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="SoftwareNumber">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  Version of the device-specific software (firmware) to
                    which the device's low-level software conforms.
      Type:         Literal
      Resolution:   Locked
      Examples:     "2"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="VideoInputEncoder">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#SoftwarePlatform"/>
                                <rdfs:comment>
      Description:  List of video input encoders supported by the device.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "MPEG-1", "MPEG-2", "H.261"
   </rdfs:comment>
                </rdf:Description>
                <!-- ***************************************************************** -->
                <!-- ***** Component: NetworkCharacteristics ***** -->

                <rdf:Description ID="SupportedBluetoothVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#NetworkCharacteristics "/>
                                <rdfs:comment>
                Description: Supported Bluetooth version.
                Type:                           Literal
                Resolution:      Locked
                Examples:        "1.0"
          </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="CurrentBearerService">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
                                <rdfs:comment>
      Description:  The bearer on which the current session was opened.
      Type:         Literal
      Resolution:   Locked
      Examples:     "OneWaySMS", "GUTS", "TwoWayPacket"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="SecuritySupport">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
                                <rdfs:comment>
      Description:  List of types of security or encryption mechanisms supported by the device.
      Type:         Literal (bag)
      Resolution:   Locked
      Example:      "WTLS-1", WTLS-2", "WTLS-3", "signText", "PPTP"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="SupportedBearers">
```

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
                                <rdfs:comment>
      Description:  List of bearers supported by the device.
      Type:         Literal (bag)
      Resolution:   Locked
      Examples:     "GPRS", "GUTS", "SMS", CSD", "USSD"
   </rdfs:comment>
                </rdf:Description>
                <!-- ***************************************************************** -->
                <!-- ***** Component: BrowserUA ***** -->

                <rdf:Description ID="BrowserName">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Name of the browser user agent associated with the
                    current request.
      Type:         Literal
      Resolution:   Locked
      Examples:     "Mozilla", "MSIE", "WAP42"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="BrowserVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Version of the browser.
      Type:         Literal
      Resolution:   Locked
      Examples:     "1.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="DownloadableBrowserApps">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  List of executable content types which the browser
                    supports and which it is willing to accept from the
                    network. The property value is a list of MIME types,
                    where each item in the list is a content type
                    descriptor as specified by RFC 2045.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "application/x-java-vm/java-applet"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="FramesCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Indicates whether the browser is capable of displaying
                    frames.
      Type:         Boolean
      Resolution:   Override
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="HtmlVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Version of HyperText Markup Language (HTML) supported
                    by the browser.
      Type:         Literal
      Resolution:   Locked
      Examples:     "2.0", "3.2", "4.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JavaAppletEnabled">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Indicates whether the browser supports Java applets.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JavaScriptEnabled">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Indicates whether the browser supports JavaScript.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="JavaScriptVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Version of the JavaScript language supported by the
                    browser.
      Type:         Literal
      Resolution:   Locked
      Examples:     "1.4"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="PreferenceForFrames">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Indicates the user's preference for receiving HTML
                    content that contains frames.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="TablesCapable">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Indicates whether the browser is capable of displaying
                    tables.
      Type:         Boolean
      Resolution:   Locked
      Examples:     "Yes", "No"
   </rdfs:comment>
                </rdf:Description>
```

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                <rdf:Description ID="XhtmlVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  Version of XHTML supported by the browser.
      Type:         Literal
      Resolution:   Locked
      Examples:     "1.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="XhtmlModules">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#BrowserUA"/>
                                <rdfs:comment>
      Description:  List of XHTML modules supported by the browser. Property
                    value is a list of module names, where each item in the
                    list is the name of an XHTML module as defined by the
                    W3C document "Modularization of XHTML", Section 4. List
                    items are separated by white space. Note that the
                    referenced document is a work in progress. Any subsequent
                    changes to the module naming conventions should be
                    reflected in the values of this property.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "XHTML1-struct", "XHTML1-blkstruct", "XHTML1-frames"
   </rdfs:comment>
                </rdf:Description>

                <!-- ************************************************************** -->
                <!-- ***** Component: WapCharacteristics ***** -->

                <rdf:Description ID="SupportedPictogramSet">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
Description: Pictogram classes supported by the device as defined in "WAP Pictogram specification".
        Type:   Literal (bag)
        Resolution:  Append
        Examples:               "core", "core/operation", "human"
  </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WapDeviceClass">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  Classification of the device based on capabilities as
                    identified in the WAP 1.1 specifications. Current
                    values are "A", "B" and "C".
      Type:         Literal
      Resolution:   Locked
      Examples:     "A"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WapVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  Version of WAP supported.
      Type:         Literal
      Resolution:   Locked
      Examples:     "1.1", "1.2.1", "2.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WmlDeckSize">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  Maximum size of a WML deck that can be downloaded to
                    the device. This may be an estimate of the maximum size
                    if the true maximum size is not known. Value is number
                    of bytes.
      Type:         Number
      Resolution:   Locked
      Examples:     "4096"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WmlScriptLibraries">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  List of mandatory and optional libraries supported in
                    the device's WMLScript VM.
      Type:         Literal (bag)
      Resolution:   Locked
      Examples:     "Lang", "Float", "String", "URL", "WMLBrowser", "Dialogs", "PSTOR"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WmlScriptVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  List of WMLScript versions supported by the device.
                    Property value is a list of version numbers, where
                    each item in the list is a version string conforming
                    to Version.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "1.1", "1.2"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WmlVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  List of WML language versions supported by the device.
                    Property value is a list of version numbers, where
                    each item in the list is a version string conforming
                    to Version.
      Type:         Literal (bag)
      Resolution:   Append
      Examples:     "1.1", "2.0"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WtaiLibraries">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  List of WTAI network common and network specific
                    libraries supported by the device.
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

```
                    Property value is a list of WTA library names, where each
                    item in the list list is a library name as specified by
                    "WAP WTAI" and its addendums. Any future addendums to "WAP WTAI" should be
                    reflected in the values of this property.
      Type:         Literal (bag)
      Resolution:   Locked
      Examples:     "WTAVoiceCall", "WTANetText", "WTAPhoneBook",
                    "WTACallLog", "WTAMisc", "WTAGSM", "WTAIS136", "WTAPDC"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="WtaVersion">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#WapCharacteristics"/>
                                <rdfs:comment>
      Description:  Version of WTA user agent.
      Type:         Literal
      Resolution:   Locked
      Examples:     "1.1"
   </rdfs:comment>
                </rdf:Description>
                <!-- ****************************************************************** -->
                <!-- ***** Component: PushCharacteristics ***** -->

                <rdf:Description ID="Push-Accept">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#PushCharacteristics"/>
                                <rdfs:comment>
      Description:  List of content types the device supports, which can be carried
                    inside the message/http entity body when OTA-HTTP is used.
                    Property value is a list of MIME types, where each item in the
                    list is a content type descriptor as specified by RFC 2045.
      Type:         Literal (bag)
      Resolution:   Override
      Examples:     "text/html", "text/plain", "image/gif"
   </rdfs:comment>
                </rdf:Description>


                   <rdf:Description ID="Push-Accept-Charset">
                      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
                      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                      <rdfs:domain rdf:resource="#PushCharacteristics"/>
                      <rdfs:comment>
                         Description:  List of character sets the device supports. Property
                                       value is a list of character sets, where each item in
                                       the list is a character set name registered with IANA.
                         Type:         Literal (bag)
                         Resolution:   Override
                         Examples:     "US-ASCII", "ISO-8859-1", "Shift_JIS"
                      </rdfs:comment>
                   </rdf:Description>

                   <rdf:Description ID="Push-Accept-Encoding">
                      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
                      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                      <rdfs:domain rdf:resource="#PushCharacteristics"/>
                      <rdfs:comment>
                         Description:  List of transfer encodings the device supports.
                                       Property value is a list of transfer encodings, where
                                       each item in the list is a transfer encoding name as
                                       specified by RFC 2045 and registered with IANA.
                         Type:         Literal (bag)
                         Resolution:   Override
                         Examples:     "base64", "quoted-printable"
                     </rdfs:comment>
                   </rdf:Description>

                   <rdf:Description ID="Push-Accept-Language">
```

```
                       <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
                       <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Seq"/>
                       <rdfs:domain rdf:resource="#PushCharacteristics"/>
                       <rdfs:comment>
                          Description:  List of preferred document languages. If a resource is
                                        available in more than one natural language, the server
                                        can use this property to determine which version of the
                                        resource to send to the device. The first item in the
                                        list should be considered the user's first choice, the
                                        second the second choice, and so on. Property value is
                                        a list of natural languages, where each item in the list
                                        is the name of a language as defined by RFC 3066[RFC3066].
                          Type:         Literal (sequence)
                          Resolution:   Override
                          Examples:     "zh-CN", "en", "fr"
                       </rdfs:comment>
   </rdf:Description>

                <rdf:Description ID="Push-Accept-AppID">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Bag"/>
                                <rdfs:domain rdf:resource="#PushCharacteristics"/>
                                <rdfs:comment>
      Description:  List of applications the device supports, where each item
                    in the list is an application-id on absoluteURI format
                    as specified in [PushMsg]. A wildcard ("*") may be used
                    to indicate support for any application.
      Type:         Literal (bag)
      Resolution:   Override
      Examples:     "x-wap-application:wml.ua", "*"
  </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="Push-MsgSize">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#PushCharacteristics"/>
                                <rdfs:comment>
      Description:  Maximum size of a push message that the device can
                    handle. Value is number of bytes.
      Type:         Number
      Resolution:   Override
      Examples:     "1024", "1400"
   </rdfs:comment>
                </rdf:Description>

                <rdf:Description ID="Push-MaxPushReq">
                                <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Property"/>
                                <rdfs:domain rdf:resource="#PushCharacteristics"/>
                                <rdfs:comment>
      Description:  Maximum number of outstanding push requests that the device
                    can handle.
      Type:         Number
      Resolution:   Override
      Examples:     "1", "5"
   </rdfs:comment>
                </rdf:Description>
</rdf:RDF>
```

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Appendix B. Summary of User Agent Profile Schema

This section is informative.

The table below summarizes the components and attributes defined within the WAP User Agent Profile schema (see Appendix A). The tag names put in parenthesis are deprecated.

Attribute	Description	Resolution Rule	Type	Sample Values
Component: HardwarePlatform				
BluetoothProfile	Supported Bluetooth profiles as defined in the Bluetooth specification [BLT]	Locked	Literal (bag)	"dialup", "lanAccess"
BitsPerPixel	The number of bits of color or grayscale information per pixel	Override	Number	"2", "8"
ColorCapable	Whether the device display supports color	Override	Boolean	"Yes", "No"
CPU	Name and model number of device CPU	Locked	Literal	"Pentium III", "PowerPC 750"
ImageCapable	Whether the device supports the display of images	Locked	Boolean	"Yes", "No"
InputCharSet	List of character sets supported by the device for text entry	Append	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"
Keyboard	Type of keyboard supported by the device	Locked	Literal	"Disambiguating", "Qwerty", "PhoneKeypad"
Model	Model number assigned to the terminal device by the vendor or manufacturer	Locked	Literal	"Mustang GT", "Q30"
NumberOfSoftKeys	Number of soft keys available on the device.	Locked	Number	"3", "2"
OutputCharSet	List of character sets supported by the device for output to the display	Append	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"
PixelAspectRatio	Ratio of pixel width to pixel height	Locked	Dimension	"1x2"
PointingResolution	Type of resolution of the pointing accessory supported by the device	Locked	Literal	"Character", "Line", "Pixel"
ScreenSize	The size of the device's screen in units of pixels	Locked	Dimension	"160x160", "640x480"

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

ScreenSizeChar	Size of the device's screen in units of characters. (Number of characters per row)x(Number of rows). In calculating this attribute use the largest character in the device's default font.	Locked	Dimension	"12x4", "16x8"
SoundOutputCapable	Indicates whether the device supports sound output	Locked	Boolean	"Yes", "No"
StandardFontProportional	Indicates whether the device's standard font is proportional	Locked	Boolean	"Yes", "No"
TextInputCapable	Indicates whether the device supports alpha-numeric text entry	Locked	Boolean	"Yes", "No"
Vendor	Name of the vendor manufacturing the terminal device	Locked	Literal	"Ford", "Lexus"
VoiceInputCapable	Indicates whether the device supports any form of voice input, including speech recognition	Locked	Boolean	"Yes", "No"
Component: SoftwarePlatform				
AcceptDownloadableSoftware	Indicates the user's preference on whether to accept downloadable software	Locked	Boolean	"Yes", "No"
AudioInputEncoder	List of audio input encoders supported by the device	Append	Literal (bag)	"G.711"
CcppAccept	List of content types the device supports	Append	Literal (bag)	"text/html", "text/plain", "text/html", "image/gif"
CcppAccept-Charset	List of character sets the device supports	Append	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"
CcppAccept-Encoding	List of transfer encodings the device supports	Append	Literal (bag)	"base64", "quoted-printable"
CcppAccept-Language	List of preferred document languages	Append	Literal (sequence)	"zh-CN", "en", "fr"

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

DownloadableSoftwareSupport	List of executable content types which the device supports and which it is willing to accept from the network	Locked	Literal (bag)	"application/x-msdos-exe"
JavaEnabled	Indicates whether the device supports a Java virtual machine	Locked	Boolean	"Yes", "No"
JavaPlatform	The list of JAVA platforms and profiles installed in the device.	Append	Literal (bag)	"Pjava/1.1.3-compatible", "MIDP/1.0-compatible", "J2SE/1.0-compatible"
JVMVersion	List of the Java virtual machines installed on the device	Append	Literal (bag)	"SunJRE/1.2", "MSJVM/1.0"
(MexeClassmark)	ETSI MexE classmark	Locked	Number	"1", "2"
MexeSpec	Class mark specialization	Locked	Literal	"7.02"
MexeClassmarks	List of MExE classmarks supported by the device.	Locked	Literal (bag)	"1", "3"
MexeSecureDomains	Indicates whether the device supports MExE security domains as specified in the MExE specifications	Locked	Boolean	"Yes", "No"
OSName	Name of the device's operating system	Locked	Literal	"Mac OS", "Windows NT"
OSVendor	Vendor of the device's operating system	Locked	Literal	"Apple", "Microsoft"
OSVersion	Version of the device's operating system	Locked	Literal	"6.0", "4.5"
RecipientAppAgent	User agent associated with the current request	Locked	Literal	"BrowserMail"
SoftwareNumber	Version of the device-specific software (firmware) to which the device's low-level software conforms	Locked	Literal	"2"
VideoInputEncoder	List of video input encoders supported by the device	Append	Literal (bag)	"MPEG-1", "MPEG-2", "H.261"
Component: NetworkCharacteristics				
CurrentBearerService	The bearer on which the current session was opened	Locked	Literal	"OneWaySMS", "GUTS", "TwoWayPacket"

© 2001, Wireless Application Protocol Forum, Ltd. All rights reserved

SecuritySupport	List of types of security or encryption mechanisms supported by the device.	Locked	Literal (bag)	"WTLS-1", WTLS-2", "WTLS-3", "signText", "PPTP"
SupportedBearers	List of bearers supported by the device	Locked	Literal (bag)	"GPRS", "GUTS", "SMS", CSD", "USSD"
SupportedBluetooth Version	Supported Bluetooth version	Locked	Literal	"1.0"
Component: BrowserUA				
BrowserName	Name of the browser user agent associated with the current request	Locked	Literal	"Mozilla", "MSIE", "WAP42"
BrowserVersion	Version of the browser	Locked	Literal	"1.0"
DownloadableBrowse rApps	List of executable content types which the browser supports and which it is willing to accept from the network	Append	Literal (bag)	"application/x-java-vm/java-applet"
FramesCapable	Indicates whether the browser is capable of displaying frames	Override	Boolean	"Yes", "No"
HtmlVersion	Version of HyperText Markup Language (HTML) supported by the browser	Locked	Literal	"2.0", "3.2", "4.0"
JavaAppletEnabled	Indicates whether the browser supports Java applets	Locked	Boolean	"Yes", "No"
JavaScriptEnabled	Indicates whether the browser supports JavaScript	Locked	Boolean	"Yes", "No"
JavaScriptVersion	Version of the JavaScript language supported by the browser	Locked	Literal	"1.4"
PreferenceForFrame s	Indicates the user's preference for receiving HTML content that contains frames	Locked	Boolean	"Yes", "No"
TablesCapable	Indicates whether the browser is capable of displaying tables	Locked	Boolean	"Yes", "No"
XhtmlVersion	Version of XHTML supported by the browser	Locked	Literal	"1.0"
XhtmlModules	List of XHTML modules supported by the browser	Append	Literal (bag)	"XHTML1-struct", "XHTML1-blkstruct", "XHTML1-frames"

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Component: WapCharacteristics				
SupportedPictogramSet	Pictogram classes supported by the device as defined in the WAP Pictogram specification	Append	Literal (bag)	"core", "core/operation", "human"
WapDeviceClass	Classification of the device based on capabilities as identified in the WAP 1.1 specifications	Locked	Literal	"A"
(WapPushMsgPriority)	User's preference on the priority of incoming push messages	Locked	Literal	"critical", "low", "none"
(WapPushMsgSize)	Maximum size of a push message that the device can handle	Locked	Number	"1024"
WapVersion	Version of WAP supported	Locked	Literal	"1.1", "1.2.1", "2.0"
WmlDeckSize	Maximum size of a WML deck that can be downloaded to the device	Locked	Number	"4096"
WmlScriptLibraries	List of mandatory and optional libraries supported in the device's WMLScript VM	Locked	Literal (bag)	"Lang", "Float", "String", "URL", "WMLBrowser", "Dialogs", "PSTOR"
WmlScriptVersion	List of WMLScript versions supported by the device	Append	Literal (bag)	"1.1", "1.2"
WmlVersion	List of WML language versions supported by the device	Append	Literal (bag)	"1.1", "2.0"
WtaiLibraries	List of WTAI network common and network specific libraries supported by the device.	Locked	Literal (bag)	"WTAVoiceCall", "WTANetText", "WTAPhoneBook", "WTACallLog", "WTAMisc", "WTAGSM", "WTAIS136", "WTAPDC"
WtaVersion	Version of WTA user agent	Locked	Literal	"1.1"
(WapSupportedApplications)	List of applications supported by the WAP device that are accessible using the push application addressing framework. Each value is a URI and represents an application identifier which may be reigstered with WINA.	Locked	Literal (bag)	“urn:x-wap-application:push.sia”, “urn:x-wap-application:wml.ua”,

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Component: PushCharacteristics				
Push-Accept	List of content types the device supports for push	Override	Literal (bag)	"text/html", "text/plain", "image/gif"
Push-Accept-Charset	List of character sets the device supports for push	Override	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"
Push-Accept-Encoding	List of transfer encodings the device supports for push	Override	Literal (bag)	"base64", "quoted-printable"
Push-Accept-Language	List of preferred document languages for push	Override	Literal (sequence)	"zh-CN", "en", "fr"
Push-Accept-AppID	List of applications the device supports for push	Override	Literal (bag)	"x-wap-application:wml.ua", "*"
Push-MsgSize	Maximum size in bytes of a push message that the device can handle	Override	Number	"1024", "1400"
Push-MaxPushReq	Maximum number of outstanding push requests that the device can handle	Override	Number	"1", "5"

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Appendix C. Reserved Attributes

This section is normative.

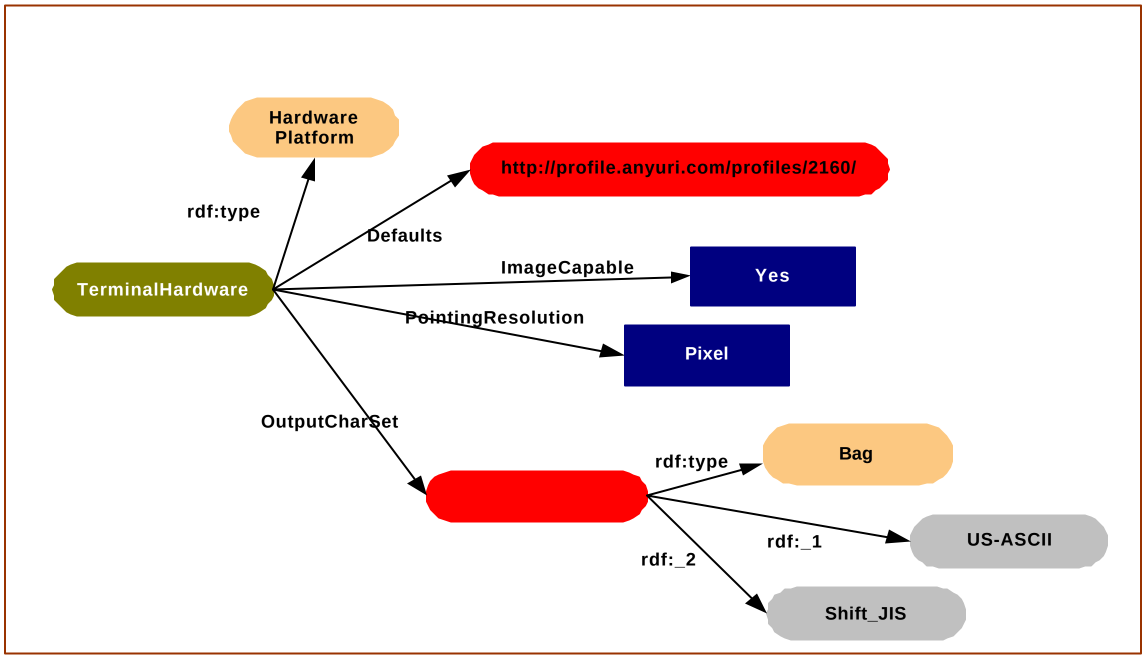
The following attributes are not currently included in the base vocabulary for User Agent Profiles, but are reserved for possible incorporation at a future date. Extended vocabularies SHOULD NOT make use of these attribute names.

```
<rdf:Description ID="LocationCapable">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
   <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
   <rdfs:comment>
      Description:  Indicates whether the device supports determination and
                    transmission of its location.
      Type:         Boolean
      Resolution:   Override
      Examples:     "Yes", "No"
   </rdfs:comment>
</rdf:Description>

<rdf:Description ID="LocationEncoding">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
   <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
   <rdfs:comment>
      Description:  Description of the encoding of the device's location.
      Type:         Literal
      Resolution:   Locked
      Examples:     "lat-long", "cell-id"
   </rdfs:comment>
</rdf:Description>

<rdf:Description ID="LocationValue">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
   <rdfs:domain rdf:resource="#NetworkCharacteristics"/>
   <rdfs:comment>
      Description:  Description of the device's location.
      Type:         Literal
      Resolution:   Override
   </rdfs:comment>
</rdf:Description>
```

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Appendix D. Serialized and Abbreviated Syntaxes

This section provides a comparison of the long and abbreviated syntax of XML encoded RDF using an example indicated in the following RDF graph (verified with SiRPAC [SiRPAC]). This section is informative.



1) The corresponding RDF syntax:

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#">
   <rdf:Description ID="TerminalHardware">
      <rdf:type resource="http://www.wapforum.org/profiles/UAPROF/ccppschema-
20010430#HardwarePlatform"/>
      <prf:defaults rdf:resource="http://profile.anyuri.com/profiles/2160"/>
      <prf:OutputCharset>
         <rdf:Bag>
            <rdf:li>US-ASCII</rdf:li>
            <rdf:li>Shift_JIS</rdf:li>
         </rdf:Bag>
      </prf:OutputCharset>
      <prf:ImageCapable>Yes</prf:ImageCapable>
      <prf:PointingResolution>pixel</prf:PointingResolution>
   </rdf:Description>
</RDF>
```

2) The equivalent compact or abbreviated syntax:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!ENTITY a 'http://profile.anyuri.com/profiles/'>
    <!ENTITY b 'http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#'>
    <!ENTITY c 'file:profile_example2.rdf#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:a="&a;"
    xmlns:b="&b;"
    xmlns:c="&c;">
<b:HardwarePlatform rdf:about="&c;TerminalHardware"
    b:ImageCapable="Yes"
    b:PointingResolution="pixel">
   <b:OutputCharset rdf:resource="&c;genid2"/>
   <b:defaults rdf:resource="&a;2160"/>
</b:HardwarePlatform>
<rdf:Bag rdf:about="&c;genid2"
    rdf:_1="US-ASCII"
    rdf:_2="Shift_JIS"/>
</rdf:RDF>
```

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Appendix E. Static Conformance Requirements (Normative)

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

1.1 Client

1.1.1 Binary encoding

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-B-C-001	Profile encoding	8	O	

1.1.2 Protocol

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-proto-C-001	Profile transport	Section 9	M	WAG-UA-Proto-C-002 OR WAG-UA-Proto-C-003
WAG-UA-proto-C-002	Profile transport using WSP	Section 9.2	O	WAG-UA-P-C-001 AND WAG-UA-P-C-002 AND WAG-UA-P-C-003 AND WAG-UA-P-C-004 AND WAG-UA-P-C-006
WAG-UA-proto-C-003	Profile transport using W-HTTP	Section 9.1	O	WAG-UA-P-C-009

Transport Variant : WSP

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-P-C-001	Profile header in WSP Connect	9.2.1.2	O	WAG-UA-P-C-006 AND WSP-CO-C-001
WAG-UA-P-C-002	Behaviour on missing Profile-warning in response		O	
WAG-UA-P-C-003	Order of request and session headers	9.2.3.2	O	
WAG-UA-P-C-004	Does the client comply with the rules in [14] and [16]?	9.2.3	O	
WAG-UA-P-C-005	Additonal headers with requests	9.2.3.3	O	
WAG-UA-P-C-006	Does the Profile header match the	9.2.2.1	O	

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

	production rule [1]?			
WAG-UA-P-C-007	Does the Profile-Diff header match the production rule [4]?	9.2.2.2	O	
WAG-UA-P-C-008	Does the Profile-warning header match the production rule [8]?	9.2.2.3	O	

© 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Transport Variant : W-HTTP

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-P-C-009	Client generates x-wap-profile header	9.1.1.1	O	
WAG-UA-P-C -010	Client generates x-wap-profile-diff header	9.1.1.2	O	WAG-UA-P-C-012 AND WAG-UA-P-C-013
WAG-UA-P-C-011	Client processes x-wap-profile-warning header	9.1.1.3	O	
WAG-UA-P-C-012	x-wap-profile contains sequnence number anf digest of instances x-wap-profile-diff	9.1.2.2	O	
WAG-UA-P-C-013	Corrupt x-wap-profile-diff header values will be ignored.	9.1.2.2	O	

1.2 Gateway

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-proto-S-001	Profile transport	Section 9	M	WAG-UA-proto-S-002 OR WAG-UA-Proto-S-003
WAG-UA-proto-S-002	Profile transport using WSP	Section9.2	O	WAG-UA-P-S-001 AND WAG-UA-P-S-002 AND WAG-UA-P-S-003 AND WAG-UA-P-S-004 AND WAG-UA-P-S-005 AND WAG-UA-P-S-006 AND WAG-UA-P-S-010
WAG-UA-proto-S-003	Profile transport using W-HTTP	Section9.1	O	WAG-UA-P-S-011 AND WAG-UA-P-S-012 AND WAG-UA-P-S-013

1.2.1 Transport Variant WSP

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-P-S-001	Does the gateway comply with the rules in [15]?	9.2.3	O	WSP:MSF
WAG-UA-P-S-002	Does the gateway comply with the rules in [17]?	9.2.3.2	O	

© 2001, Wireless Application Protocol Forum, Ltd. All rights reserved

WAG-UA-P-S-003	Does the gateway comply with the rules in [18]?	9.2.3.2	O	
WAG-UA-P-S-004	Does the gateway comply with the rules in [19]?	9.2.3.3	O	
WAG-UA-P-S-005	Does the gateway comply with the rules in [20]?	9.2.3.3	O	
WAG-UA-P-S-006	Does the gateway comply with the rules in [21]?	9.2.3.3	O	
WAG-UA-P-S-007	Caching of Profile and Profile-Diff headers	9.1.2	O	WSP:MSF
WAG-UA-P-S-009	Transmission of multiple profiles in one header	9.1.3	O	
WAG-UA-P-S-010	Decoding of the Profile section in Profile-Diff headers	9.1.3	O	WBXML

1.2.2 Transport Variant W-HTTP

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-P-S-011	Does the gateway pass x-wap-profile header through	9.1.1	O	
WAG-UA-P-S-012	Doess the gatway pass x-wap-profile-diff header through	9.1.1	O	
WAG-UA-P-S-013	Does the gateway pass x-wap-profile-warning header through	9.1.1	O	

1.3 Schemas

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-Schema-001	Is the schema a valid WAP User Agent Profile Schema?		M	WAG-UA-S-002 AND WAG-UA-S-003 AND WAG-UA-S-004 AND WAG-UA-S-005 AND WAG-UA-S-006 AND WAG-UA-S-008 AND

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

				WAG-UA-S-009 AND WAG-UA-S-010 AND WAG-UA-S-011 AND WAG-UA-S-012 AND WAG-UA-SA-002 AND WAG-UA-SA-003 AND WAG-UA-SA-005 AND WAG-UA-SA-006 AND WAG-UA-SA-007 AND WAG-UA-SA-008 AND WAG-UA-SA-010 AND WAG-UA-SP-002 AND WAG-UA-SP-003 AND WAG-UA-SP-004 AND WAG-UA-SP-005 AND WAG-UA-SP-006 AND WAG-UA-SX-001 AND WAG-UA-SX-002 AND WAG-UA-SX-003 AND WAG-UA-SX-004 AND WAG-UA-SX-005

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-S-001	Does the schema correspond to a vocabulary specified in conformity with the RDF standard?	7.1	O	
WAG-UA-S-002	Is the namespace of the schema identified using the prefixes “rdf” and “rdfs”?	7.1	M	
WAG-UA-S-003	Are all namespace declarations in the schema declared as attributes of the	7.1	M	

	"rdf:RDF" root element?			
WAG-UA-S-004	Are all namespace aliases defined outside of nested XML elements of the schema?	7.1	M	
WAG-UA-S-005	Does a unique URI serve as an unambiguous identifier for the vocabulary?	7.1	O	
WAG-UA-S-006	Does the schema consist of at least one component?	7.1	O	
WAG-UA-S-007	Does each component in said schema describe a set of attributes within one or more description blocks?	7.1	O	
WAG-UA-S-008	Do all components in the schema have the same schema structure (layout)?	7.1	O	
WAG-UA-S-009	Is every component in the schema an object of the type Class?	7.1	O	
WAG-UA-S-010	Does each component contain a subordinate description block for default attributes?	7.1	O	
WAG-UA-S-011	Are default attributes described within the Default description block?	7.1	O	
WAG-UA-S-0012	Are descriptions to override default values included in the component description, but outside the Default description block?	7.1	O	

1.3.1 Attributes

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-SA-001	Does each profile attribute belong to only one component in the schema?	7.3	O	
WAG-UA-SA-002	Is a domain constraint (rdfs:domain) used to describe the attribute-component relationship in the schema?	7.3	O	
WAG-UA-SA-003	Is each profile attribute defined using a name and value pair syntax?	7.3	O	
WAG-UA-SA-005	Is every attribute description within an RDF description embedded inline to denote the value, or expressed as an RDF resource?	7.3	O	
WAG-UA-SA-006	Are all multi value and composite attributes described as RDF	7.3	O	

Ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

	resources?			
WAG-UA-SA-007	For all cases when lists of values are associated with given attributes, are RDF containers (Bag or sequence) used?	7.3	O	
WAG-UA-SA-008	Is the description of the attribute within the Default tag resolved first? Do other descriptions of the attribute override the default description? Is the ultimate value of the attribute determined by the resolution rules for that attribute? (see section 7.3 (*Locked*, *Override*, *Append*))	7.3	O	
WAG-UA-SA-010	Does the attribute follow the naming conventions defined in RDF, and the RDF schema?	7.3	O	

1.3.2 Profile

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-SP-001	Is the root node identified with an invariant node ID?	7.4	O	
WAG-UA-SP-002	Does the profile contain one or more attributes identified in the base vocabulary?	7.4	O	
WAG-UA-SP-003	Are the component instances of the components identified in the schema?	7.4	O	
WAG-UA-SP-004	Do all profile parsers parse the profile based on the component type?	7.4	O	
WAG-UA-SP-005	Is the relative order of the profiles to be merged maintained?	7.5	M	
WAG-UA-SP-006	Are default property values loaded first?	7.5	M	

1.3.3 XML Namespace

Item	Function	Reference	Mandatory / Optional	Requirement
WAG-UA-SX-001	Does the WBXML encoder convert the document's chosen prefix to the prefix defined for that namespace in table 8.1, when it encounters a namespace	8.2.1	O	

	declaration that matches one of the allowed namespaces and its prefix does not match the defined prefix.			
WAG-UA-SX-002	Does the encoder, when it encounters a namespace other than those defined here, encode all elements from that namespace using literal tags?	8.2.1	O	
WAG-UA-SX-003	Does the encoder, when it encounters a namespace other than those defined here, encode all attribute names for elements defined in that namespace using literal names?	8.2.1	O	
WAG-UA-SX-004	Does the encoder, when it encounters a namespace other than those defined here, preserve the namespace prefix in the encoding of the tags and attribute names?	8.2.1	O	
WAG-UA-SX-005	Does the encoder, when it encounters a namespace other than those defined here, preserve all namespace declarations (instances of the attribute xmlns)?	8.2.1	O	

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved

Appendix F. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	0-Oct-2001		The initial version of this document.
Class 2	20-Oct	7, 8, 9, E	Updates after public review.

ó 2001, Wireless Application Protocol Forum, Ltd.
All rights reserved