WAP International Pictogram Specification

Draft Version 09-July-2000

Wireless Application Protocol

Pictogram Specification

Version 0.10

Disclaimer:

This document subject to change without notice.

Contents

1. Scope

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

This specification defines the common pictogram set and its architecture. The common pictogram set is a set of pictograms that the user agents (e.g., mobile phones) may internally have their images, and content authors may use these images in the contents without increasing network traffics. On the other hand, manufactures may install images of pictograms that were appropriate for the device (e.g., size, colour, image format, etc.).

To meet the requirement of the worldwide market, the common pictogram set is classified into several classes. For example, some pictograms have operational, culture and time independent semantics and intended to be used generally. The other pictograms are glossary of symbols that represent certain embodiments. Such pictograms are classified into different classes.

The bundles of images could be added to the device as a collection of pictogram in a class. The image bundles could be factory installed, network operators updated, or end user controlled. Images of pictograms could be stored into permanent or temporary storage of the device.

Some user agents that are not capable of displaying images may have alternative way to present pictograms: e.g., character only device may display alternative text instead of its image.

This specification defines a set of semantics of pictogram. However glyphs of pictograms, which represent the image the pictogram may have when they are rendered or displayed, are out of scope of this document.

2. Document Status

This document is available online in the following formats:

- PDF format at http://www.wapforum.org/.

2.1 Copyright Notice

2.2 Errata

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

2.3 Comments

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

3. References

3.1 Normative References

[WAEOVER] "Wireless Application Environment Overview", WAP Forum, 04-November-1999. URL: http://www.wapforum.org/

[WBXML] "WAP Binary XML Content Format", WAP Forum, 04-November-1999. URL: http://www.wapforum.org/

[WML] "Wireless Markup Language Specification", WAP Forum, 04-November-1999. URL: http://www.wapforum.org/

[WMLScript] "WMLScript Specification", WAP Forum, 04-November-1999. URL: http://www.wapforum.org/

3.2 Informative References

[CSS2] “Cascading Style Sheets, level 2 CSS2 Specification”, W3C Recommendation 12-May-1998, URL; http://www.w3.org/TR/REC-CSS2

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, et al., August 1998. URL: http://www.ietf.org/rfc/rfc2396.txt.

[SVG] “Scalable Vector Graphics (SVG) 1.0 Specification”, W3C Working Draft 03 December 1999. URL: http://www.w3.org/TR/SVG

[UNICODE] "The Unicode Standard: Version 2.0", The Unicode Consortium, Addison-Wesley Developers Press, 1996. URL: http://www.unicode.org/

[XML] "Extensible Markup Language (XML), W3C Proposed Recommendation 10-February-1998, REC-xml-19980210", T. Bray, et al, February 10, 1998. URL: http://www.w3.org/TR/REC-xml

4. Definitions and abbreviations

All non-trivial abbreviations and definitions used in this document are listed in the following sections. The definitions section includes description of general concepts and issues that may be fully defined in other documents. The purpose of this section is merely to advise the reader on the terminology used in the document.

4.1 Definitions

The notation used in the specification part of this document uses the common elements defined here.

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].

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

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

Common Pictogram Set - a set of common pictograms defined in this document.

Core Pictogram Set - a set of pictograms that is supported by all device classes. The core pictogram set is the subset of the common pictogram set.

Content - synonym for data objects.

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

Content Format - actual representation of content.

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

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

WML - The Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrowband device, e.g., a phone.

WMLScript - A scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript™ scripting language.

XML - the Extensible Markup Language is a World Wide Web Consortium (W3C) proposed standard for Internet markup languages, of which WML is one such language. XML is a restricted subset of SGML.

4.2 Abbreviations

The following abbreviations apply to this document.

CGI	Common Gateway Interface
CSS	Cascading Style Sheet [CSS2]
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
RFC	Request For Comments
SVG	Scalable Vector Graphic [SVG]
URI	Uniform Resource Identifier [RFC2396]
URL	Uniform Resource Locator [RFC2396]
W3C	World Wide Web Consortium
WSP	Wireless Session Protocol
WAP	Wireless Application Protocol
WAE	Wireless Application Environment
WBMP	Wireless BitMaP
XML	Extensible Markup Language

5. Architectural Overview

5.1 Technical Approaches

This section is informative.

A pictogram is an icon-like image that is rendered within the text, and shows more intuitive information than text. A user agent MUST render a pictogram to fit with following and/or preceding characters. This document defines the common pictogram set in semantics, but not image or glyph. The user agent SHOULD render a pictogram in the most appropriate way for the device capabilities. E.g., The user agent that support color display may render a pictogram as colored image, the other user agent may render the pictogram as an animation image, and a text only user agent should render alternative text instead of the image.

This specification provides an extensible mechanism that makes it possible to add a new pictogram to the common pictogram set in the future. Currently, pictogram mechanism is defined to be used only with the WML, however, migration to the other markup language such as next version of WML is also considered.

Pictograms are usually pre-installed by the device manufactures, and by using the pictogram mechanism, authors can reduce content data size that is downloaded over the network than using ordinary inline graphical images. It is also possible to install new pictograms over the network.

As pictograms have characteristics of both characters and graphical images, there are two approaches to the pictogram mechanism.

- Pictograms as Characters

  Unicode already includes a number of pictogram symbols, and it is a considerable to register common pictogram set in this specification with Unicode. Pictograms in Unicode may be used in any content where Unicode characters are legal. Such pictograms may be used not only in WML but also in WMLScript or even in the plain text. A Pictogram in Unicode encoded in UTF-8 or UTF-16 costs 2 to 4 bytes, and it is the most efficient way in reducing the content data size. However, colored or animated pictograms are not allowed in this approach. It is also difficult to register a new pictogram with Unicode in the future; that means this mechanism provides less extensibility.

  However, several technologies as SVC [SVG] defined in W3C has a mechanism to use non-Unicode characters in the content. CSS2 [CSS2] also provides a mechanism to download the font data over the network. Such mechanisms may improve extensibility.

- Pictograms as Graphical Images

  In contrast, graphical image pictogram approach enables it very easy to add a new pictogram to the set; it is also possible to use rich presentation feature such as color or animation, since a pictogram is treated as a graphical image. However, syntax to present image pictogram requires relatively larger than Unicode pictogram encoded in UTF-8 or UTF-16. It is also difficult to use graphical image pictogram in the content other than markup language.

As both approaches have advantages and disadvantages, both character and image approach should coexist. However, this document focuses on specifying graphical image pictogram mechanism in this version. Some pictograms in the common pictogram set may be considered as candidates to the next version of Unicode, however, it is out of scope of this document. Extensible character approach, such as SVG mechanism with CSS2 downloadable font will be discussed in the future.

5.2 Pictogram Class

This section is normative.

To meet the requirement of the worldwide market, the common pictogram set is classified into several classes.

Pictograms that have operational, culture and time independent semantics and intended to be used generally are classified in the core class. All the user agents must support pictograms in the core class.

The other pictograms that are glossary of symbols that represent certain embodiments are classified into several classes that represent reasonable sets of pictograms. In this document, such classes are called Pictogram Dictionary. Pictograms in the Pictogram Dictionary classes may depend of certain region, culture or time. Such pictograms are classified into different classes.

Pictogram sets in the following classes are defined in this document.

TBD.

Category of the Class	***Class Name***	***Description***
Core	core	The Core Pictogram set that has operational semantics and culture independent.
Pictogram Dictionary	human/gender	
	human/body	
	human/humanlike	
	animal	
	plant	
	weather	
	horoscope	

Note: List up all the classes including Core. This will be a normative info.
Note:
How should we define classes of Pictograms?
Alternative 1) To define only core pictogram set in this spec, and make operators (or manufactures, content providers) to define pictogram dictionaries they want.
Pros: Operators that already have pictograms can use them without any changes to the list.
Cons: Duplicated pictograms will be defined by each operator and causes inefficient use of memory on the device.
Alternative 2) To define the core pictogram set and pictogram dictionaries as defined in trial pictogram set (approximately 200 pictograms). In addition, to provide extensible mechanism for the pictograms.
Pros: We can share common pictograms not only in the core pictogram set but pictogram dictionaries.
Cons: Does the trial set have problems in internationalization point of view?
In i18n-WG meeting, we agreed to recommend alternative 2 as consensus of the group.

5.3 Installation of the Pictograms

This section is normative.

Device manufactures MAY install certain sets of pictograms and their images into the permanent storage of the device. At least, the core pictogram set MUST be installed by the device manufactures.

When the pictogram is presented in the content, it always has its local name and URL of its resource (e.g., graphical image data of the pictogram). If the user agent supports graphical image but does not have the image of the pictogram, it SHOULD download the image from the specified URL over the network. The user agent SHOULD store the resource to re-use when the pictogram with same local name is used in the same or in the other content.

There may be a couple of mechanisms to install pictograms over the network. However, in this version of the pictogram specification, only pull mechanism is defined. Other mechanisms such as push will be discussed in the future, but not guaranteed.

A user agent MAY provide a means to allow end-users to install, update, or remove pictograms with such as personal pictogram editor feature. However, such kind of mechanism is out of scope of this document. Pictograms installed by end users SHOULD override already installed pictograms that have the same local name.

5.4 Pictogram Name

This section is normative.

Each pictogram is identified by its local name. The syntax of local name is restricted to the subset of URI syntax defined in [RFC2396]. Missing production rule in the following BNF depends on [RFC2396].

```
pict_URI      = abs_URI | rel_URI
abs_URI       = scheme ":" net_path
rel_URI       = ( net_path | abs_path | rel_path )
scheme        = "pict"
net_path      = "//" authority abs_path
abs_path      = "/" path_segments
rel_path      = pictogram_name

authority     = server
server        = hostport
hostport      = host [ ":" port ]
host          = hostname | IPv4address
hostname      = *( domainlabel "." ) toplabel [ "." ]
domainlabel   = alphanum | alphanum *( alphanum | "-" ) alphanum
toplabel      = alpha | alpha *( alphanum | "-" ) alphanum
IPv4address   = 1*digit "." 1*digit "." 1*digit "." 1*digit
port          = *digit

path_segments = class_name "/" pictogram_name
class_name    = segment *( "/" segment )
segment       = *pchar
pictogram_name= *pchar
pchar         = unreserved | escaped |
                ":" | "@" | "&" | "=" | "+" | "$" | ","
```

Manufactures or authors that attempt to add a pictogram class MUST use their own `authority` to keep uniqueness in the URI. The `authority` of "`www.wapforum.org`" is reserved for the common pictogram set.
`class_name` is the name of class of the pictogram defined in Section 5.1 or defined by manufactures or authors.
`pictogram_name` is the name of the pictogram in the class.

Scheme (i.e. pict), `authority`, and `class_name` MAY be omitted. A user agent MUST assume that the default value of the `authority` is "`www.wapforum.org`", and default of the `class_name` is "`core`". E.g., following four local names are identical.

```
pict://www.wapforum.org/core/rightArrow
//www.wapforum.org/core/rightArrow
/core/rightArror
rightArrow
```

If the authority is not default, the local name must be in the absolute URN syntax. e.g.,

```
pict://www.foo.com/sports/jp/judo
```

Note that although the local name has URI syntax, it does not indicate the location of the pictogram, but the unique identifier of the pictogram in the hierarchical naming manner.

5.5 Alternative Presentation

This section is normative.

Some user agents that are not capable of displaying images may have alternative way to present pictograms: e.g., character only device may display alternative text instead of the graphical image.

The alternative presentation MUST be specified in the content.

Alternative presentation may vary on type of user agent. For example, user agent with character only display MAY render an alternative string, and voice browser MAY pronounce it.

5.6 Capability Negotiation of Pictogram

Note: Add Use cases. – reduce data size, etc. Need more use cases?

This section is normative.

A user agent MAY negotiate with origin servers or gateways about which sets of pictograms it supports. Origin servers or gateway MAY compose the content to suit the capability of the user agent. E.g., origin server or gateway MAY replace unsupported pictograms to alternative characters to reduce overall data size.
When the capability negotiation is carried out, UAProf with the `SupportedPictogramSet` attribute is used for capability negotiation of pictogram as defined in [UAProf].
Class name of pictogram MUST be specified as an attribute value.

6. Pictogram Presentation on WAE User Agents

This section is normative.

6.1 Pictogram in WML

Pictograms MUST be represented using the `img` element and the localsrc attribute in WML.
The following example illustrates how to specify a right arrow pictogram in the WML deck.

```
<img localsrc="pict:core/rightArrow"
    src="http://www.pict.com/xx/rightArrow.wbmp"
    alt="-&gt;"/>
```

In the `img` element, the `localsrc` attribute has local name of the pictogram. The `src` attribute has URI of the resource of the pictogram, and *alt* attribute has alternative text.
Above example shall be rendered as ⇨ (in image supported device) or " `->`" (when the image is not supported)
Graphical image capable user agents MUST display a pictogram that is in the core pictogram set as an image. If a user agent does not support graphical images, it MUST present pictogram in the alternative way (e.g., render alternative text).

6.2 Pictogram in WBXML

When a WML deck is encoded into binary format, pictogram information in the deck MUST be encoded into short binary format to reduce network traffic. A list of attribute start tokens and attribute values are defined in [7 Pictogram Set].

6.3 General Processing Model

A conformant WAE user agent MUST process the pictogram in the following manner.

1. If the content is encoded in the WBXML format, decode it.
2. An `img` element, which has `localsrc` attribute, MUST be treated as a pictogram; the `localsrc` attribute value is its local name, the `src` attribute value is the URL of its resource, and the `alt` attribute value is its alternative text.
3. If the authority is omitted in the local name, the user agent MUST assume that it is `"www.wapforum.org"`. If the class name of the pictogram is omitted in the local name, the user agent MUST assume that it is `"core"`.
4. If the user agent supports graphical image and has image data of the pictogram, it MUST render the image to suit proceeding and following characters.
5. If the user agent supports graphical image and does not have the image data of the pictogram, it SHOULD download the image data from the specified URL and render it to suit preceding and following characters.
6. If the user agent does not support graphical image, or unable to download the image by any reason, it MUST render or present the alternative text in the most appropriate way to its capability.

7. Pictogram Set

This section is normative.

This specification defines a set of semantics of pictogram. However glyphs of pictograms, which represent the image the pictogram may have when they are rendered or displayed, are out of scope of this document.

7.1 The Core Pictogram Set

Category	Name	Description	Binary Token Value
arrow	upArrow	up arrow	
	downArrow	down arrow	
	rightArrow	right arrow	
	leftArror	left arrow	
	upperRightArrow		
	upperLeftArrow		
	lowerRightArrow		
	lowerLeftArrow		
	fingerUp	Pointing finger, up	
	fingerDown	Pointing finger, down	
	fingerRight	Pointing finger, right	
	fingerLeft	Pointing finger, left	
buttons	button1		
	button2		
	button3		
	button4		
	button5		
	button6		
	button7		
	button8		
	button9		
	button0		
operation buttons	makePhoneCall		
	find		
	userAuthentication		
	password		
	nextPage		
	clear		
	stop		
	TOP		
	NEXT		
	BACK		
message operation	receiveMessage		
	sendMessage		
	message	mail/envelope	
	document	document	
	attachement	attachement/paper clip	
	folder	folder	
	inbox	inbox	

Category	Name	Description	Binary Token Value
	outbox	outbox	
state	secure	Secure	
	insecure	Non secure	
	copyright		
	trademark		
	underConstruction		
	beginner		
... and so on.			

Note:
We will define list of pictogram name, description (and WBXML token value) of the core pictogram set here.

7.2 Pictogram Dictionary

Note:
We will define list of pictogram name, description (and WBXML token value) of the pictogram dictionaries here.

Appendix A: Static Conformance Requirement

This static conformance requirement defines a minimum set of features that can be implemented to ensure that WAE User Agents and WAE Servers will be able to inter-operate. While both WAE User Agent behavior and WAE server behavior are described in the WAP Pictogram Specification, not all items apply to both entities, so there are separate tables for each.
A feature can be optional, mandatory.

1. WAE User Agent

1.1 Pictogram Architecture

Item	Function	Reference	Status
WPCT-CA-001	Core class	5.2, 7.1	M
WPCT-CA-002	Pictogram Dictionary	5.2, 7.2	O
WPCT-CA-003	Manufacture installation of core pictogram set	5.3	M
WPCT-CA-004	Manufacture installation of other pictogram set	5.3	O
WPCT-CA-005	Network installation by pull	5.3	O
WPCT-CA-006	End-user installation	5.3	O
WPCT-CA-007	pictogram URI	5.4	M
WPCT-CA-008	alternative presentation	5.5	M
WPCT-CA-009	capability negothiation	5.6	O

1.2 Pictogram in WAE

Item	Function	Reference	Status
WPCT-CW-001	Pictogram in WML	6.1	M
WPCT-CW-002	Pictogram in WBXML	6.2	M
WPCT-CW-003	General Processing Model	6.3	M

2. Static Conformance Requirement – WAE Server

2.1 Pictogram Architecture

Item	Function	Reference	Status
WPCT-SA-001	capability negothiation	5.6	O

2.2 Pictogram in WAE

Item	Function	Reference	Status
WPCT-SW-001	WBXML encoding of pictograms	6.2	M