Wireless CSS Specification

Approved Version 1.2 – 29 Mar 2011

Open Mobile Alliance

OMA-TS-WCSS-V1_2-20110329-A

Contents

1. Scope

This section is informative.

Wireless CSS specifies a subset of CSS2.1 [[CSS21**Error! Reference source not found.**] and mobile extensions defined by OMA. It can be used to style XHTML Mobile Profile [[XHTMLMP] documents, as well as any other XML document.

The CSS2 subset contains core CSS functionality such as inheritance, cascading, selectors, and the syntax. It also contains properties for the box layout model, text, fonts, visual effects, colour, and lists.

The following CSS extensions are specified:

- **Marquee** - properties to create simple animation effects. Typically this will be used to create scrolling text.
- **Input – (**Obsoleted**)** - properties to specify the format of the user input into a form control. It is the same feature as the WML ‘format’ and ‘emptyok’ attributes provide.
- **Accesskey – (**Obsolete**)** - property to specify an optional, additional way to activate an element in a document. It is similar to the feature the HTML ‘accesskey’ attribute provides.

A subset of CSS2 is also specified in the W3C, the “W3C CSS Mobile Profile 2.0” Candidate Recommendation [[CSSMP]. Since both CSS2 variants follow CSS user agent semantics, conforming Wireless CSS user agents will accept valid W3C CSS Mobile Profile style sheets.

2. References

2.1 Normative References

[CSS21] "Cascading Style Sheets, level 2 revision 1", CSS2.1 Specification, W3C.Candidate Recommendation, 19 July 2007
URL: http://www.w3.org/TR/CSS21/

[CSS3BOX] "CSS Box Module", W3C Working Draft, 9 March 2007, URL: http://www.w3.org/Style/Group/css3-src/css3-box /

[CSS3NS] "CSS Namespace Module", W3C Working Draft 1 October 2006, URL:
http://www.w3.org/Style/Group/css3-src/css3-namespace/

[CSSMP20] "CSS Mobile Profile 2.0", W3C Candidate Recommendation URL:
http://www.w3.org/Style/Group/css3-src/css-mobile/

[CDFWICD] "WICD Mobile 1.0", W3C Candidate Recommendation, 18 July 2007, URL:
http://www.w3.org/TR/WICDMobile/

[IOPProc] "OMA Interoperability Policy and Process". Open Mobile Alliance™. OMA-IOP-Process-v1_4,.
URL:http//www.openmobilealliance.org/

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

[RFC2318] "The text/css Media Type", H. Lie, B. Bos, C. Lilley, March 1998.
URL: *http://www.ietf.org/rfc/rfc2318.txt*

[WCSS11] "Wireless CSS Specification 1.1", 20 October 2006, Open Mobile Alliance™. , OMA-WAP-WCSS-V1_1-20061020-A.pdf

[XMLCSS] "Associating Style Sheets with XML documents", Version 1.0, W3C Recommendation.
URL: *http://www.w3.org/1999/06/REC-xml-stylesheet-19990629*

Editor's Note:Need to update some W3C URLs to the "public" versions when available

2.2 Informative References

[BASIC] "XHTML Basic 1.1 ", W3C Candidate Recommendation, 13 July 2007
URL: http://www.w3.org/TR/xhtml-basic/

[HTML40] ""HTML 4.01 Specification", W3C Recommendation 24 December 1999,
http://www.w3.org/TR/1999/REC-html401-19991224

[XHTMLMP] "XHTML Mobile Profile 1.3", Open Mobile Alliance™. OMA-TS-XHTMLMP-V1_3. URL: http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [[RFC2119].

CSS property, descriptor, and pseudo-class names are delimited by single quotes.

CSS values are delimited by double quotes.

Data types are written as

<data type>

Notes are written as

Note: An informational message to the reader.

3.1.1 Tables

CSS properties and features are defined in tables with the following column names:

- **PROPERTY**
  Comma-separated list of property names.
- **FEATURE**
  Name of a CSS feature.
- **VALUE**
  “|”-separated list of datatypes and property values. (This is not the syntax of the property value.)
- **REFERENCE**
  Normative reference to the place where the property or function is defined.
- **STATUS**
  “M” – Mandatory; “O” – Optional.

3.1.2 Obsolete Properties

For user agent developers, properties that have been designated as obsolete SHOULD NOT be built into new implementations. Older implementations MAY still support these properties. However, all implementations MUST be able to parse, and at a minimum ignore obsoleted syntax.

Content providers are discouraged from using obsolete properties.

3.2 Abbreviations

CSS	Cascading Style Sheets
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WCSS	Wireless Cascading Style Sheets
WML	Wireless Markup Language
XHTML	Extensible Hyper Text Markup Language
XML	Extensible Markup Language

4. Introduction

This section is informative.

Readers of this specification are assumed to be familiar with Cascading Style Sheets (CSS) as defined in [[CSS21**Error! Reference source not found.**].

4.1 How to Read This Specification

The major part of this specification is a table with normative references to the CSS2.1 [[CSS21] specification. Appendix A follows the syntax described in [IOPProc] and is the Static Conformance Requirements (SCR) for the whole specification.

It is recommended that the specification is read in the following way:

1. Read Appendix A to get an overall view of what parts of the specification are mandatory and what parts are optional. The SCR is a list of features of Wireless CSS.

2. For each feature in the SCR, details on conformance can be found in the referenced section.

A reference to CSS includes the referenced section and all sub-sections, unless stated otherwise.

4.2 Conformance

Conformance to this specification is defined in the Static Conformance Requirements (SCR) table in Appendix A. A conforming user agent may support additional CSS features.

This specification does not specify user interface presentation. It is not specified what a dotted line or a solid line should look like or how colors should be represented on the display. The look and feel depends on the device. This limits to what extent certain features can be tested. It is however expected that conformance testing will be carried out in such a way that it is possible to distinguish between a device that has implemented a particular feature and one that has not. It can be done for example by observing just the difference in effect of two different property values. A device that is unable to support a certain feature cannot claim to conform to that feature.

> **Note**: Conformance to this specification is intended to be stronger than for CSS2.1, where conformance can be claimed even if parts of the specification have not been implemented (see [[CSS21**Error! Reference source not found.**] section 3.2). In Wireless CSS conformance can be claimed, in the Implementation Conformance Statement (ICS), only for features that have actually been implemented.

Fortunately, in many cases it is clear how a property value should be presented. Whether a font is “sans serif” or whether the text is indented five pixels can be tested. And if that test fails, the device does not conform to that feature.

Some features are optional and some features are mandatory. Here are some of the principles that have been used to determine what shall be mandatory and what shall be optional:

- For interoperability, most of the core CSS features are mandatory: syntax, selectors, cascading, inheritance, how values are calculated, etc.

- For consistency, when there are variants of the same property, such as ‘border-left’ and ‘border-right’ of the ‘border’ property, then all variants are required.

- In general, property values that are keywords, such as “larger”, “left”, and “smaller”, are required for that property.

- When a property takes values of only one type, that type is required.

- Some properties can take many different types of values. A property requires at least one type of values. The requirement-column in the SCR table is used to express the dependency between a property and the types of values it can take.

- Unimplemented property values (ones that are listed as 'OPTIONAL' and not implemented or found in [[CSS21] but not explicitly listed in this specification) MUST still be parsed and handled correctly as unsupported values.
- Values expressed as colours always require support for the 17 CSS 2.1 colour keywords (16 HTML 4.0 colours [HTML40] + orange) and the RGB specification.
- Properties may be OPTIONAL if they are required for the support of technologies external to Browsing 2.4. These will be noted both in text and in the SCR Appendix. (e.g. support for CDF WICD [CDFWICD]). Properties that are optional except when supporting CDF WICD are noted as "O OR CDFWICD:M"

4.2.1 Exceptions when Presenting Documents in Certain Character Sets

There are style sheet properties that are not useful for documents in a certain character set. Most properties can be applied to a document in a Latin character set. Some properties for fonts and text, however, cannot be applied in a meaningful way to a document using Chinese-Japanese-Korean (CJK) character sets.

The general principle is that properties that are not useful for a certain character set are optional when a document in that character set is presented.

In the requirements tables, properties which are not useful when CJK documents are presented, are marked as optional ("M OR CJK:O"). A user agent presenting a CJK document is not required to implement these properties, although they are mandatory when presenting a document in another character set.

In this specification, exceptions to the conformance requirements are made for CJK documents. Future versions may take other character sets into account, in addition to CJK.

4.2.2 Shorthand Properties

Shorthand properties allow authors to specify the value of several properties using a single property. It is a syntactical shortcut. All shorthand properties are mandatory. Since some of the properties which can be specified by the shorthand property may be optional – and not supported by the user agent – the properties which actually can be set depend on the user agent.

4.2.3 Colour and "Black & White" Devices

Devices may be characterized as having one or more bits of color depth. Devices that have only one bit of color depth (assumed black and white for the purposes of this definition) should treat all color requests other than white as black, except where both foreground color and background color are specified and they are different. In this case the color with the higher hexadecimal value should default to white, and the color with the lower hexadecimal value should default to black. This prevents text being rendered invisible unintentionally.

4.3 The 'text/css' Media Type

Since Wireless CSS style sheets are either valid CSS2.1 style sheets [[CSS21**Error! Reference source not found.**] or contain extensions expressed in valid CSS syntax, when delivered over the Internet, the 'text/css' [[RFC2318] MIME media type can be used to represent Wireless CSS style sheets.

OMA has not defined a MIME media type specifically for Wireless CSS style sheets.

5. Selectors

This section is normative.

5.1 Pattern Matching

In order to support the Wireless CSS Selectors the user agent MUST implement all mandatory items (“M”) in the following table.

FUNCTION	REFERENCE	STATUS
Universal selector	[[CSSMP]Sec. 3, [[CSS21] section 5.3	M
Type selectors	[[CSSMP]Sec. 3, [[CSS21] section 5.4	M
Descendant selectors	[[CSSMP]Sec. 3, [[CSS21] section 5.5	M
Child selectors	[[CSSMP]Sec. 3, [[CSS21] section 5.6	M
Link pseudo-class: :link	[[CSSMP]Sec. 3, [[CSS21] section 5.11.2	M
Link pseudo-class: :visited	[[CSSMP]Sec. 3, [[CSS21] section 5.11.2	M
Dynamic pseudo-class: :active	[[CSSMP]Sec. 3, [[CSS21] section 5.11.3	M
Dynamic pseudo-class: :focus	[[CSSMP]Sec. 3, [[CSS21] section 5.11.3	M
Class selectors	[[CSSMP]Sec. 3, [[CSS21] section 5.8.3	M
ID selectors	[[CSSMP]Sec. 3, [[CSS21] section 5.9	M
Grouping	[[CSSMP]Sec. 3, [[CSS21] section 5.2.1	M

6. Syntax and Basic Data Types

This section is normative.

6.1 Syntax and Parsing

In order to support the Wireless CSS Syntax and Parsing the user agent MUST implement all mandatory items (“M”) in the following table.

FUNCTION	REFERENCE	STATUS
CSS style sheet syntax	[[CSSMP]Sec. 6, [[CSS21] section 4.1	M
Rules for handling parsing errors	[[[CSS21] section 4.2	M
Algorithm for determining the character encoding	[[CSS21] section 4.4	M
The “charset” at-rule in external style sheets	[[CSSMP]Sec. 4, [[CSS21] section 4.4	M
The “namespace” at-rule	[[CSSMP]Sec. 3, [[CSS3NSSec. 3	O OR CDFWICD: M

6.2 Data Types

In order to support the Wireless CSS Data Types the user agent MUST implement all mandatory items (“M”) in the following table.

FUNCTION	REFERENCE	STATUS
<number> Integers and real numbers.	[[CSSMP]Sec. 6, [[CSS21] section 4.3.1	M
<length> The “px”, “em”, and “ex” length units.	[[CSSMP]Sec. 6, [[CSS21] section 4.3.2	M
<percentage>	[[CSSMP]Sec. 6, [[CSS21] section 4.3.3	M
<uri>	[[CSSMP]Sec. 6, [[CSS21] section 4.3.4	M
<color> The 17 HTML 4.0 colours.	[[CSSMP]Sec. 6, [[CSS21] section 4.3.6	M
<color> Numerical RGB specification.	[[CSSMP]Sec. 6, [[CSS21] section 4.3.6	M
<string>	[[CSSMP]Sec. 6, [[CSS21] section 4.3.7	M

7. Assigning Property Values, Cascading, and Inheritance

This section is normative.

In order to support the Wireless CSS Assigning property values, cascading, and inheritance the user agent MUST implement all mandatory items (“M”) in the following table.

FUNCTION	REFERENCE	STATUS
Specified, computed, and actual values	[[CSS21] section 6.1	M
Inheritance	[[CSS21] section 6.2,	M
The “import” at-rule	[[CSSMP]Sec. 4, [[CSS21] section 6.3	M
The cascade	[[CSS21] section 6.4	M

Note: [[CSS21**Error! Reference source not found.**] section 6.4.4 specifies how non-CSS presentational hints (e.g. the presentational markup found in WML and XHTML) are used in the CSS cascade. It also specifies the same thing for the style rules in the XHTML “style” attribute.

8. Media Types

This section is normative.

In order to support the Wireless CSS Media Types the user agent MUST implement all mandatory items (“M”) in the following table.

FUNCTION	VALUE	REFERENCE	STATUS
@media	handheld	[[CSSMP]Sec. 4, [[CSS21] section 7.2.1, 7.3	M
	all	[[CSSMP]Sec. 4, [[CSS21] section 7.2.1,7.3	M

Note: A user agent MAY support additional media types, such as “aural” if it supports aural style sheets.

9. Associating Style Sheets with XML Document

This section is normative.

In order to support the Wireless CSS Associating style sheets with XML documents the user agent MUST implement the mandatory item (“M”) in the following table.

FUNCTION	REFERENCE	STATUS
XML style sheet processing instruction	[XMLCSS]	M

Here is an (informative) example of how the XML style sheet processing instruction is used in an XHTML document:

```
<?xml version=“1.0” ?>
<?xml-stylesheet href=“mobilestyle.css” media=“handheld” type=“text/css” ?>
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML Basic 1.1//EN”
“http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd”>
<html>
  <head><title>Welcome to The Mobile Browsing Experience</title></head>
  <body>
   ...
  </body>
</html>
```

Authors SHOULD use the “handheld” media type to indicate that the style sheet is appropriate for handheld devices.

10.Box Model

This section is normative.

10.1 Margin

In order to support the Wireless CSS Margin properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
margin-top, margin-right, margin-bottom, margin-left	<margin-width> | inherit	[[CSS21] section 8.3	M
margin	<margin-width> | inherit	[[CSS21] section 8.3	M

The <margin-width> datatype is defined in [[CSS21] section 8.3.

10.2 Padding

In order to support the Wireless CSS Padding properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
padding-top, padding-right, padding-bottom, padding-left	<padding-width> | inherit	[[CSS21] section 8.4	M
padding	<padding-width> | inherit	[[CSS21] section 8.4	M

The <padding-width> datatype is defined in [[CSS21] section 8.4.

10.3 Border Width

In order to support the Wireless CSS Border Width properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
border-top-width, border-right-width, border-bottom-width, border-left-width	<border-width> | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.1	M
border-width	<border-width> | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.1	M

The <border-width> datatype is defined in [[CSS21] section 8.5.1.

10.4 Border Colour

In order to support the Wireless CSS Border Colour properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
border-top-color, border-right-color, border-bottom-color, border-left-color	<color> | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.2	M
border-color	<color> | transparent | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.2	M

10.5 Border Style

In order to support the Wireless CSS Border Style properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
border-top-style, border-right-style, border-bottom-style, border-left-style	none | solid | dashed | dotted | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.3	M
	hidden | double, groove | ridge | inset | outset	[[CSSMP] section 5, [[CSS21] section 8.5.3	O
border-style	none | solid | dashed | dotted | inherit	[[CSSMP] section 5, [[CSS21] section 8.5.3	M
	hidden | double, groove | ridge | inset | outset	[[CSSMP] section 5, [[CSS21] section 8.5.3	O

10.6 Border Shorthand Properties

In order to support the Wireless CSS Border Shorthand properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	REFERENCE	STATUS
border-top, border-right, border-bottom, border-left, border	[[CSSMP] section 5, [[CSS21] section 8.5.4	M

11.Colours and Background

This section is normative.

11.1 Foreground Colour

In order to support the Wireless CSS Foreground Colour properties the user agent MUST implement the mandatory item ("M") in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
color	<color> | inherit	[[CSSMP] section 5, [[CSS21] section 14.1	M

11.2 Background Colour

In order to support the Wireless CSS Background Colour properties the user agent MUST implement all mandatory items ("M") in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
background-color	<color> | transparent | inherit	[[CSSMP] section 5, [[CSS21] section 14.2	M

11.3 Background Images

In order to support the Wireless CSS Background Images properties the user agent MUST implement all mandatory items ("M") in the following table and Background Colour.

PROPERTY	VALUE	REFERENCE	STATUS
background-image	<uri> | none | inherit	[[CSSMP] section 5, [[CSS21] section 14.2	M
background-repeat	repeat | repeat-x | repeat-y | no-repeat | inherit	[[CSSMP] section 5, [[CSS21] section 14.2	M
background-attachment	scroll | fixed | inherit	[[CSSMP] section 5, [[CSS21] section 14.2	M
background-position	top | center | bottom | left | right | inherit	[[CSSMP] section 5, [[CSS21] section 14.2	M
	<length> | <percentage>	[[CSSMP] section 5, [[CSS21] section 14.2	O

11.4 Background Shorthand Property

In order to support the Wireless CSS Background Shorthand properties the user agent MUST implement the mandatory item ("M") in the following table.

PROPERTY	REFERENCE	STATUS
background	[[CSSMP] section 5, [[CSS21] section 14.2	M

12.Fonts

This section is normative.

12.1 Font Family

In order to support the Wireless CSS Font Family properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
font-family	<generic-family> | inherit	[[CSS21] section 15.3	M OR CJK:O
	<family-name>	[[CSS21] section 15.3	O

The <generic-family> and <family-name> datatypes are defined in [[CSS21] section 15.3.

Note: As a user agent is not required to have any particular font family installed, the mandatory nature of ‘font-family’ means that the user agent MUST process the family request and provide what it considers a best fit font from those installed.

12.2 Font Style

In order to support the Wireless CSS Font Style properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
font-style	normal | italic | oblique | inherit	[[CSSMP] section 5, [[CSS21] section 15.4	M OR CJK:O

The “oblique” font style MAY be rendered as “italic”, according to [[CSS21**Error! Reference source not found.**] section 15.4.

12.3 Font Variant

In order to support the Wireless CSS Font Variant properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
font-variant	normal | small-caps | inherit	[[CSSMP] section 5, [[CSS21] section 15.5	M OR CJK:O

12.4 Font Weight

In order to support the Wireless CSS Font Weight properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
font-weight	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit	[[CSSMP] section 5, [[CSS21] section 15.6	M OR CJK:O

It is reasonable to assume that many mobile devices will not support more than the two font-weights: “normal” and “bold”. Those two are defined in [[CSS21**Error! Reference source not found.**] to map to the numerical font weight values of “400” for “normal” and 700 for “bold”. It is also defined that a lower numerical font-weight must be lighter or equal to a higher numerical font-weight.

If user agent is unable to support different levels of font weights, it MUST map the keywords to the normal and bold keywords in the following way:

- Map font weight keywords “100”, “200”, “300”, “400”, and “lighter” to the same weight as “normal”.
- Map font weight keywords “500”, “600”, “700”, “800”, “900”, and “bolder” to the same weight as “bold”.

With two font weights the relative value “bolder” always equals “bold” and “lighter” always equals “normal”.

As specified in [[CSS21], only the strings “100”, “200”, “300”, “400”, “500”, “600”, “700”, “800”, “900” are legal as font weights. These weights are strings, not values.

12.5 Font Size

In order to support the Wireless CSS Font Size properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
font-size	xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | inherit	[[CSSMP] section 5, [[CSS21] section 15.7	M OR CJK:O
	<length> | <percentage>	[[CSSMP] section 5, [[CSS21] section 15.7	O

12.6 Font Shorthand Property

In order to support the Wireless CSS Font Shorthand properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	REFERENCE	STATUS
font	[[CSSMP] section 5, [[CSS21] section 15.8	M OR CJK:O

13.Lists

This section is normative.

13.1 Lists

In order to support the Wireless CSS Lists properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
list-style-type	disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none | inherit	[[CSSMP] section 5, [[CSS21] section 12.5.1	M OR CJK:O[1]
list-style-position	inside | outside | inherit	[[CSS21] section 12.5.1	O
list-style-image	<uri> | none | inherit	[[CSSMP] section 5, [[CSS21] section 12.5.1	M
list-style		[[CSSMP] section 5, [[CSS21] section 12.5.1	M

User agent that does not recognize a numbering system, in 'list-style-type', SHOULD use “decimal”.

Since some list properties are optional, some user agents may not support all values of the shorthand property.

The user agent behaviour when the height of the list style image is greater than the highest inline box in the same line is implementation defined. This means that to guarantee interoperability, images specified by ‘list-style-image’ should be of a height less than or equal to the inline box height in the line in which the image is used.

[1] These values do not exist in Latin1 character set, so images are used instead. However for Japanese, they exist in JIS X 0208 as characters, so images are not used.

14.Text

This section is normative.

14.1 Indentation

In order to support the Wireless CSS Text Indentation properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
text-indent	<length> | <percentage> | inherit	[[CSSMP] section 5, [[CSS21] section 16.1	M OR CJK:O

14.2 Alignment

In order to support the Wireless CSS Text Alignment properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
text-align	left | right | center | justify | inherit	[[CSSMP] section 5, [[CSS21] section 16.2	M

Note: The actual justification algorithm used depends on the user-agent and the language/script of the text.

Conforming user agents may interpret the value 'justify' as 'left' or 'right', depending on whether the element's default writing direction is left-to-right or right-to-left, respectively.

14.3 Decoration

In order to support the Wireless CSS Text Decoration properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
text-decoration	none | blink | inherit	[[CSSMP] section 5, [[CSS21] section 16.3.1	M
	underline	[[CSSMP] section 5, [[CSS21] section 16.3.1	M
	overline | line-through	[[CSSMP] section 5, [[CSS21] section 16.3.1	O

14.4 Transformation

In order to support the Wireless CSS Text Transformation properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
text-transform	capitalize | uppercase | lowercase | none | inherit	[[CSSMP] section 5, [[CSS21] section 16.5	M OR CJK:O

14.5 White Space

In order to support the Wireless CSS Text White space properties the user agent MUST implement the mandatory item (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
white-space	normal | pre | nowrap | pre-line | inherit	[[CSSMP] section 5, [[CSS21] section 16.6	M

15. Visual Effects

This section is normative.

In order to support the Wireless CSS Visual effects properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
visibility	visible | hidden | collapse | inherit	[[CSSMP] section 5, [[CSS21] section 11.2 [[CSS21] section 17.5.5	M

Note: If used on elements other than rows, row groups, columns, or column groups, 'collapse' has the same meaning as 'hidden'. The 'visibility' property takes the value 'collapse' for row, row group, column, and column group elements. This value causes the entire row or column to be removed from the display, and the space normally taken up by the row or column to be made available for other content. Contents of spanned rows and columns that intersect the collapsed column or row are clipped. The suppression of the row or column, however, does not otherwise affect the layout of the table. This allows dynamic effects to remove table rows or columns without forcing a re-layout of the table in order to account for the potential change in column constraints. [[CSS21] section 17.5.5

16. Visual Formatting

This section is normative.

16.1 Display Properties

In order to support the Wireless CSS Display properties the user agent MUST implement all mandatory items ("M") in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
display	inline | block | list-item | none | inherit	[[CSSMP] section 5, [[CSS21] section 9.2.4	M
	run-in | compact | marker | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption	[[CSSMP] section 5, [[CSS21] section 9.2.4	O

16.2 Positioning Schemes

Default for all implementations is position = static. The inclusion of these properties is to allow the positioning of entities in a compound (mixed markup) document.

PROPERTY	VALUE	REFERENCE	STATUS
position	static | relative | absolute | inherit	[CSSMP] section 5, [[CSS21] section 9.3.1	O OR CDFWICD: M
top	<relative unit> | <percentage> | auto | inherit	[CSSMP] section 5, [[CSS21] section 9.3.2	O OR CDFWICD: M
right	<relative unit> | <percentage> | auto | inherit	[CSSMP] section 5, [[CSS21] section 9.3.2	O OR CDFWICD: M
bottom	<relative unit> | <percentage> | auto | inherit	[CSSMP] section 5, [[CSS21] section 9.3.2	O OR CDFWICD: M
left	<relative unit> | <percentage> | auto | inherit	[CSSMP] section 5, [[CSS21] section 9.3.2	O OR CDFWICD: M

16.3 Float Positioning

In order to support the Wireless CSS Float Positioning properties the user agent MUST implement all mandatory items (“M”) in the following table.

PROPERTY	VALUE	REFERENCE	STATUS
float	left | right | none | inherit	[[CSSMP] section 5, [[CSS21] section 9.5.1	M

When WCSS is applied to an XHTML Mobile Profile document, the user agent MUST support the 'float' property applied to the 'img' element. A user agent MAY support applying 'float' to other elements, according to [[CSS21**Error! Reference source not found.**].

16.4 Float Flow Control

In order to support the Wireless CSS Float Flow Control properties the user agent MUST implement all mandatory items (“M”) in the following table and float positioning.

PROPERTY	VALUE	REFERENCE	STATUS
clear	left | right | both | none | inherit	[[CSSMP] section 5, [[CSS21] section 9.5.2	M

16.5 Layered Presentation

Management of layered presentation is only required when the document is a compound document and the stacking of entities using different markups causes an overlap

PROPERTY	VALUE	REFERENCE	STATUS
z-index	Auto | <integer> | inherit	[[CSSMP] section 5, [[CSS21] section 9.9	O OR CDFWICD: M

16.6 Content Width and Height

In order to support the Wireless CSS Content Width and Height properties the user agent MUST implement all mandatory items (“M”) in the following table. Note: Properties that are required for CDF WICD [CDFWICD] conformance are optional otherwise, and included to allow the control of placement of entities using different markup languages in a compound document.

PROPERTY	VALUE	REFERENCE	STATUS
width	<length> | <percentage> | auto | inherit	[[CSSMP] section 5, [[CSS21] section 10.2	M
height	<length> | <percentage> | auto | inherit	[[CSSMP] section 5, [[CSS21] section 10.5	M

PROPERTY	VALUE	REFERENCE	STATUS
min-width	<length> | <percentage> | inherit	[[CSSMP] section 5, [[CSS21] section 10.4	O OR CDFWICD: M
max-width	<length> | <percentage> | none | inherit	[[CSSMP] section 5, [[CSS21] section 10.4	O OR CDFWICD: M
min-height	<length> | <percentage> | inherit	[[CSSMP] section 5, [[CSS21] section 10.7	O OR CDFWICD: M
max-height	<length> | <percentage> | none | inherit	[[CSSMP] section 5, [[CSS21] section 10.7	O OR CDFWICD: M
vertical-align	top | middle | bottom | baseline | inherit	[[CSSMP] section 5, [[CSS21] section 10.8.1	M
	sub | super | text-top | text-bottom	[[CSS21] section 10.8.1	O
	<percentage> | <length>	[[CSS21] section 10.8.1	O

With WCSS it is possible for a content author to set the width and height of the content area, which is a CSS box, by using the 'width' and 'height' properties.

If the rendered content (see [[CSS21**Error! Reference source not found.**]) does not fit into the content area it is modified (scaled, deformed, etc.) to fit the area. The modification method is implementation defined. (In CSS2.1 the author has some control over this situation by using the 'overflow' and 'clip' properties, which is not part of WCSS.)

Some replaced elements (see [[CSS21**Error! Reference source not found.**]) such as <img> and <object> permit the content author to specify an alternative representation. If the content cannot be modified to fit into the content area, the alternative representation is used instead. For example if an image in an <img> element cannot be modified to fit into the content area, the value of the “alt” attribute is used as the rendering of the image. The alternative representation is also used in other situations, for example when the user has disabled images rendering using a user control.

An error is indicated to the user when the rendered content cannot be modified and no alternative representation is available. The user agent shall indicate the nature of the error, e.g. presentation error, resource error or content interpretation error, to the user in a meaningful way.

16.7 Outlining

Outlining properties are only mandatory when implementing compound documents [CDFWICD] and serve to highlight specific areas within a document.

PROPERTY	VALUE	REFERENCE	STATUS
outline	[<outline-color> | <outline-style> | <outline-width>] | inherit	[[CSSMP] section 5, [[CSS21] section 18.4	O OR CDFWICD: M
outline-width	<border-width> | inherit	[[CSSMP] section 5, [[CSS21] section 18.4	O OR CDFWICD: M
outline-style	none | solid | dashed | dotted | inherit	[[CSSMP] section 5, [[CSS21] section 18.4	O OR CDFWICD: M
outline-color	<color> | inherit	[[CSSMP] section 5, [[CSS21] section 18.4	O OR CDFWICD: M

17.Marquee

This section is normative.

17.1 CSS3 Marquee

PROPERTY	VALUE	REFERENCE	STATUS
overflow	auto	[CSS3BOX, Section 11.1	M
	visible | hidden | scroll	[CSS3BOX, Section 11.1	O
overflow-style	marquee	[CSS3BOX, Section 11.2	M
marquee-style	scroll | slide | alternate	[CSS3BOX, Section 11.3	M
marquee-loop	<non-negative-integer> | infinite	[CSS3BOX, Section 11.3	M
marquee-direction	forward | reverse	[CSS3BOX, Section 11.3	M
marquee-speed	slow | normal | fast	[CSS3BOX, Section 11.3	M

Note: In CSS3, marquee is a part of the overflow property. The ‘overflow’ property is assumed to have a value of ‘auto’.

Editor’s Note: This is in the private W3C area. http://www.w3.org/Style/Group/css3-src/css3-box/Overview-new.html#overflow

17.2 WAP Marquee

Support for the WAP Marquee CSS extension is obsolete.

PROPERTY	VALUE	REFERENCE	STATUS
display	-wap-marquee	[[WCSS11] Section 17.1	O
-wap-marquee-style	scroll | slide | alternate	[[WCSS11] Section 17.2	O
-wap-marquee-loop	<integer> | infinite	[[WCSS11] Section 17.3	O
-wap-marquee-dir	ltr | rtl	[[WCSS11] Section 17.4	O
-wap-marquee-speed	slow | normal | fast	[[WCSS11] Section 17.5	O

18.CSS Extension: Access Keys

This section is normative.

Support for the WAP Access Keys CSS is obsolete.

PROPERTY	VALUE	REFERENCE	STATUS
-wap-accesskey	none | <KeyCombinationList> | inherit	[[WCSS11] Section 18.1	O

19.CSS Extension: Input

This section is normative.

Support for the WAP Input CSS extensions is obsolete.

PROPERTY	VALUE	REFERENCE	STATUS
-wap-input-format	<format>	[[WCSS11] Section 19.2	O
-wap-input-required	true | false	[[WCSS11] Section 19.3	O

Appendix A. Change History (Informative)

A.1 Approved Version 1.1 History

Reference	Date	Description
OMA-WAP-WCSS-V1_1	20 Oct 2006	Approved by TP OMA Ref# OMA-TP-2006-0370R01-INP_Browsing_V2_2_for_Final_Approval.doc
WAP-239-WCSS-20011026-a	26 Oct 2001	WCSS V1.0
OMA-TS-WCSS-V1_2-20110329-A	29 Mar 2011	Status changed to Approved by TP: OMA-TP-2011-0097-INP_Browsing_V2_4_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

In order to support Wireless CSS the user agent MUST implement all mandatory items ("M") in the following table, and SHOULD implement all optional items ("O").

ITEM	FUNCTION	REFERENCE	STATUS	REQUIREMENT
WCSS-SELECTOR-C-001	Pattern matching	Section 5.1	M	
WCSS-SYNTAX-C-001	Syntax and parsing	Section 6.1	M	
WCSS-TYPES-C-001	Data types	Section 6.2	M	
WCSS-CASC-C-001	Assigning property values, cascading, and inheritance	Section 7	M	
WCSS-MEDIA-C-001	Media types	Section 8	M	
WCSS-ASOC-C-001	Associating style sheets with XML documents	Section 9	M	
WCSS-MARGIN-C-001	Margin properties	Section 10.1	M	
WCSS-PADDING-C-001	Padding properties	Section 10.2	M	
WCSS-BORDER-WIDTH-C-001	Border width	Section 10.3	M	
WCSS-BORDER-COLOR-C-001	Border colour	Section 10.4	M	
WCSS-BORDER-STYLE-C-001	Border style	Section 10.5	M	
WCSS-BORDER-SHORT-C-001	Border shorthand property	Section 10.6	M	
WCSS-FCOLOR-C-001	Foreground colour	Section 11.1	M	
WCSS-BCOLOR-C-001	Background colour	Section 11.2	M	
WCSS-BIMAGES-C-001	Background images	Section 11.3	O	
WCSS-BACKGROUND-C-001	Background shorthand property	Section 11.4	M	
WCSS-FONTS-FAMILY-C-001	Font family	Section 12.1	O	
WCSS-FONTS-STYLE-C-001	Font style	Section 12.2	M	
WCSS-FONTS-VARIANT-C-001	Font variant	Section 12.3	O	
WCSS-FONTS-WEIGHT-C-001	Font weight	Section 12.4	M	
WCSS-FONTS-SIZE-C-001	Font size	Section 12.5	O	
WCSS-FONTS-SHORT-C-001	Font shorthand property	Section 0	M	
WCSS-LISTS-C-001	Lists	Section 13.1	M	
WCSS-TEXT-INDENT-C-001	Text indentation	Section 14.1	M	

ITEM	FUNCTION	REFERENCE	STATUS	REQUIREMENT
WCSS-TEXT-ALIGN-C-001	Text alignment	Section 14.2	M	
WCSS-TEXT-DEC-C-001	Text decoration	Section 14.3	M	
WCSS-TEXT-TRANS-C-001	Text transformation	Section 14.4	O	
WCSS-TEXT-WS-C-001	White space	Section 14.5	M	
WCSS-VEFFECT-C-001	Visual effects	Section 15	O	
WCSS-VFORM-DISP-C-001	Display properties	Section 16.1	M	
WCSS-VFORM-PS-C-001	Positioning Schemes	Section 0	M	
WCSS_VFORM-FLOAT	Float Positioning	Section 16.3	M	
WCSS-VFORM-CLEAR-C-001	Float Flow Control	Section 16.4	M	
WCSS-VFORM-LAYERS-C-001	Layered Presentation	Section 16.5	O	
WCSS_VFORM-SIZE-C-001	Content Width and Height	Section 16.6	M	
WCSS_VFORM_OUTLIN-C-001	Outlining	Section 16.7	O	
WCSS-MARQUEE-C-001	Marquee	Section 17	M	
WCSS-WAPEXT-MARQUEE-C-001	CSS Extension: Marquee	Section 17	O Obsolete	
WCSS-WAPEXT-ACC-C-001	CSS Extension: Access keys	Section 18	O Obsolete	
WCSS-WAPEXT-INPUT-C-001	CSS Extension: Input	Section 19	O Obsolete	

Appendix C. Changes from WCSS 1.1 to WCSS 1.2

This section is informative.

1. The baseline specification for WCSS has been changed from CSS 2.0 to CSS 2.1 [[CSS21].
2. WCSS 1.2 now matches property for property with W3C CSS-MP 2.1.
3. Optional properties for the support of the W3C CDF WICD profile [CDFWICD] have been added.
4. All WAP extensions have been Obsoleted.
5. Marquee functionality will follow the new W3C CSS 3 Marquee
6. A bug in the display property value list has been fixed.