

WCSS Tests

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	OBJECTIVES	5
1.2	SCOPE	5
1.3	KNOWN LIMITATIONS	5
2	REFERENCES	6
3	TESTING OVERVIEW	7
3.1	TESTING METHODOLOGY	7
4	SELECTORS	7
4.1	PATTERN MATCHING	7
5	SYNTAX AND BASIC DATA TYPES	12
5.1	SYNTAX AND PARSING	12
5.2	DATA TYPES	25
6	ASSIGNING PROPERTY VALUES, CASCADING AND INHERITANCE	26
7	MEDIA TYPES	31
8	ASSOCIATING STYLE SHEETS WITH XML DOCUMENT	32
9	BOX MODEL	35
9.1	MARGIN	35
9.2	PADDING	44
9.3	BORDER WIDTH	51
9.4	BORDER COLOR	54
9.5	BORDER STYLE	56
9.6	BORDER SHORTHAND PROPERTIES	62
9.7	MORE BOX MODEL TESTS:	68
10	COLORS & BACKGROUND	74
10.1	FOREGROUND COLOR	74
10.2	BACKGROUND COLOR	76
10.3	BACKGROUND IMAGES	77
11	FONTS	89
11.1	FONT STYLE	89
11.2	FONT VARIANT	90
11.3	FONT WEIGHT	91
11.4	FONT SIZE	94
11.5	SHORTHAND FONT PROPERTY	96
11.6	FONT FAMILY	98
12	LISTS	99
12.1	LISTS	99
13	TEXT	107
13.1	INDENTATION	107
13.2	ALIGNMENT	108
13.3	DECORATION	109
13.4	TRANSFORMATION	111
13.5	WHITESPACE	113
13.6	MORE TEXT TESTS	115

14	VISUAL EFFECTS	116
15	VISUAL FORMATTING	119
15.1	DISPLAY PROPERTIES	119
15.2	FLOAT POSITIONING	121
15.3	FLOAT FLOW CONTROL	124
15.4	CONTENT WIDTH & HEIGHT	129
15.5	MORE FORMATTING TESTS	133
16	WAP CSS EXTENSION: MARQUEE	139
16.1	DISPLAY: -WAP-MARQUEE	139
16.2	MARQUEE STYLE: '-WAP-MARQUEE-STYLE'	141
16.3	MARQUEE LOOP: '-WAP-MARQUEE-LOOP'	143
16.4	MARQUEE DIRECTION: '-WAP-MARQUEE-DIR'	144
16.5	MARQUEE SPEED: '-WAP-MARQUEE-SPEED'	145
16.6	MORE MARQUEE TESTS	146
17	WAP CSS EXTENSION: ACCESS KEYS	148
17.1	ACCESS KEYS: '-WAP-ACCESSKEY'	148
17.1.1	Access Key Availability and Assignment	149
17.1.2	Fallbacks and Multiple Assignments	150
18	WAP CSS EXTENSION: INPUT	154
18.1	GENERAL CONCEPTS	154
18.2	INPUT-FORMAT: '-WAP-INPUT-FORMAT'	154
18.2.1	The WML 'format' attribute	158
18.3	REQUIRED INPUT: '-WAP-INPUT-REQUIRED'	159
18.3.1	Input Format and Input Required Combination	160
18.3.2	The WML 'emptyok' Attribute	162
19	INHERITANCE TESTS	163
19.1	MARGIN	164
19.2	PADDING	169
19.3	BORDER WIDTH	174
19.4	BORDER COLOR	179
19.5	BORDER STYLE	184
19.6	BORDER SHORTHAND PROPERTIES	193
19.7	COLORS	198
19.8	FONTS	200
19.9	LISTS	203
19.10	TEXT	205
19.11	VISUAL EFFECTS	208
19.12	VISUAL FORMATTING	209
20	TABLE TESTS	214
21	CSS2 EXTENSIONS	216
21.1	SELECTORS	216
22	CSS MP EXTENSIONS	220

1 INTRODUCTION

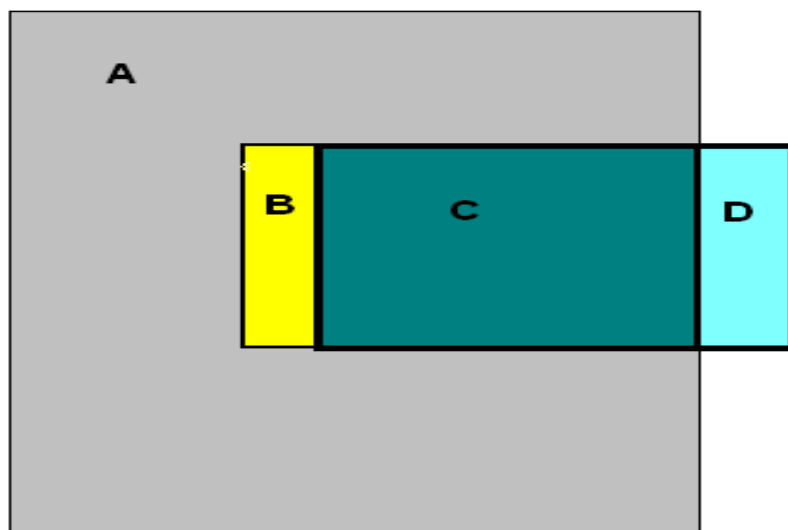
This test plan establishes the guidelines for testing the user-agent's support to the *WAP Cascading Style Sheets* specification [WAPCSS] using WAPCSS test suite. The intention of this testing is to find out the extent of user-agent's support of this specification, but not to verify the strict conformance to the specification as the functionality of individual user-agents vary greatly depending on the capabilities and features of the devices.

Style sheets allow the web content developers to separate the presentation properties of documents from their content. These style sheets can be attached to the structured documents (eg: XHTML) and delivered to the target media. The style sheets use the style sheet language defined by W3C's *Cascading Style Sheets, level 2* specification [CSS2].

Both of the *WAP Cascading Style Sheets* specification [WAPCSS] and *CSS Mobile Profile* [CSSMP] specification are subsets of same [CSS2] specification and are tailored to the constraints and needs of the mobile devices. The [WAPCSS] extends a subset of [CSS2] with a few WAP specific extensions (marquee, input format and accesskey) whereas [CSSMP] is totally a subset of [CSS2] and has no additional extensions over [CSS2]. But [CSSMP] specifies five additional absolute length data types of [CSS2] which are not part of [WAPCSS].

These [WAPCSS] and [CSSMP] specifications share the same core [CSS2] functionality such as syntax, data types, selectors, cascading and inheritance and most of the properties and values. As the core language of these two specifications is same, the [WAPCSS] supporting user-agent must accept valid [CSSMP] style sheet also. Further, a [WAPCSS] supporting user-agent may or may not support the additional properties and values specified by [CSSMP] or [CSS2], but when such content is encountered, it must handle them gracefully.

The three specifications [CSS2], [WAPCSS] and [CSSMP] can be represented schematically as shown in the following diagram.



In the above diagram, the area **A+B+C** represents [CSS2]. The area **B+C** represents [CSSMP]. The area **C+D** represents [WAPCSS].

The test content covered in Sections 5 to 16 and Sections 20 and 21 of this document is represented by area **C**, which is part of all three specifications. The test content of Sections 17

to 19 is specific to [WAPCSS] only and represented by area **D**. The area **B** represents content which is not part of [WAPCSS], but part of both [CSS2] and [CSSMP] is covered in Section 23. Finally the area **A** represents content specific to [CSS2] only and is partly covered in Section 22.

1.1 Objectives

- To verify that the WAPCSS functionality supported by the Browser is as per [WAPCSS] specification.
- To verify the capability of the Browser to handle additional valid content of [CSSMP] without panic or crashing.
- To verify the capability of the Browser to handle additional valid content of [CSS2] without panic or crashing.
- To verify that the Browser can handle any invalid content without panic or crashing.

1.2 Scope

Currently, all tests within the WAPCSS test suite are manual. There are 358 tests within this test suite in which about 50 are negative tests. Depending on the test engineer's familiarity of the tests, device used and the frequency of failures, the estimated time to run all the tests would be approximately 20 hours. This estimation is excluding the time required to log the failures observed to the defect-tracking database.

1.3 Known Limitations

Section 23 of this test plan describes the test content for additional features of [CSSMP] over [WAPCSS]. At present, the [CSSMP] is still a W3C's candidate recommendation and subject to change. The test content of Section 23 may be changed in future, if there are any changes made to [CSSMP].

2 REFERENCES

- [WAPCSS] **WAP Cascading Style Sheets Specification**
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-239-WCSS-20011026-a.pdf>
- [CSS2] **Cascading Style Sheets, level 2, CSS2 Specification**,
W3C Recommendation, 19 December 2000
<http://www.w3.org/TR/1998/REC-CSS2-19980512>
- [CSSMP] **CSS Mobile Profile**,
W3C Candidate Recommendation, 25 July 2002
<http://www.w3.org/TR/css-mobile>
- [XMLPI] **Associating Style Sheets with XML documents, v1.0**
W3C Recommendation, 29 June 1999
<http://www.w3.org/TR/xml-stylesheet>
- [XHTMLBasic] **XHTML Basic**, W3C Recommendation 19 December 2000
<http://www.w3.org/TR/xhtml-basic/>
- [WML13] **WML 1.3**, WAP-238-WML,
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-191-WML-20000219-a.pdf>
- [XHTMLMP] **XHTML Mobile Profile**, 29 Oct, 2001
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-277-XHTMLMP-20011029-a.pdf>

3 TESTING OVERVIEW

3.1 Testing Methodology

Sections 5 to 22 of this document describe the actual tests. Each test description contains the test case number, test case title, and the physical file name with path to be used *relative to the root directory of the test suite*. One or more applicable assertions and their traceability to the specifications are specified with each test case description. Finally the verification to be carried out is described. Wherever necessary, an example of expected display is also given for clarity.

Typical test contains a single XHTML page (either XHTML Basic [XHTMLBasic] or XHTML Mobile Profile [XHTMLMP] page) associated with style sheets. The style sheets are associated with XHTML page using any of the external, internal and inline style sheet methods or a combination of these methods.

Load the page into the user-agent and verify that the content is rendered correctly as described in this document. For the tests, which uses unsupported features or invalid content, the display is not important as far as browser is loading the page without crashing. For example, consider a test in which the display of background images has to be verified. But, if the UA does not support the background images feature, then it is not a bug if the page loads without error, but the background images are not displayed. But, if the UA supports background images feature and in the test they don't show up, then it is an error and report it as a defect.

Note that for each test given, the actual page as seen on the screen of the browser may only give a summary or partial list of verification actions for the tester to complete. This is done intentionally due to screen considerations and to limit the scrolling on the user-agent. A complete list of verifications is always detailed in this test plan, and therefore the test plan should be carefully consulted for each test case before posting a result. All the tests need manual verification.

4 SELECTORS

4.1 Pattern Matching

Each test in this section refers to SCR: WCSS-SELECTOR-C-001. The tests are described below (45.14.01 to 45.14.12). Further, section 22.1 contains tests for the additional selectors of [CSS2] which can be used as unsupported content tests for [WAPCSS] user-agent.

<u>Test #</u>	<u>Test Name</u>	<u>Starting Page</u>
45.14.01	Universal Selector	wcss_selectors/sel_universal.xhtml

Assertion: When the Universal Selector is used, the rule must be applied to any single element in the document tree.

Traceability:

[WAPCSS]	Section 5.1
[CSS2]	Section 5.3

Verification: Verify that entire text in the page is displayed in bold letters only.

45.14.02 **Type Selector**

wcss_selectors/sel_type.xhtml

Assertion: When the Type Selector is used, the rule must be applied to every instance of the element type in the document tree.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.4

Verification: Verify that, only the text in "PARA-1" is displayed in all bold letters.

45.14.03 **Descendant A B**

wcss_selectors/sel_des1.xhtml

Assertion: When the Descendant Selector of the form " A B" is used, the rule must be applied to every instance of the B element, which is an arbitrary descendant of some ancestor element A.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.5

Verification: Verify that the words "direct child", "Grand child" and "Great grand child" are displayed in all underlined. They may also appear as emphasized or strong text.

45.14.04 Descendant A * B

wcss_selectors/sel_des2.xhtml

Assertion: When the Descendant Selector of the form " A * B" is used, the rule must be applied to every instance of the B element which is a grandchild or later descendant of element A.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.5

Verification: Verify that the words "Grand child" and "Great grand child" appear in underlined and emphasized letters. Verify that the word "direct child" appears in emphasized letters and is not underlined.

45.14.05 Pseudo-class: link

wcss_selectors/sel_ps_link.xhtml

Assertion: When the Link pseudo-class: link is used, the rule must be applied to all links in the document.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.11.2

Verification: Verify that all un-visited hyperlinks in this page are displayed in italic text only. Activate the un-visited links. Then navigate back to this page and verify that the just visited link now appears in normal case.

45.14.06 Other Pseudo-classes

wcss_selectors/sel_ps_others.xhtml

Assertion 1: When the Link pseudo-class: visited is used, the rule must be applied to all links in the document, which have been visited.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.11.2

Assertion 2 : When the Dynamic pseudo-class: active is used, the rule must be applied while the user is activating an element.

Assertion 3 : When the Dynamic pseudo-class: focus is used, the rule must be applied while an element has the focus.

Traceability of above 2 assertions:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.11.3

Verification: Verify that all visited links are displayed in lower case text only. Verify that active link appears without underline and the link in focus appears in blinking text.

45.14.07 Class Selector

wcss_selectors/sel_class.xhtml

Assertion: When the Class Selector is used, the rule must be applied to all elements matching the class attributes in the document tree.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.8.3

Verification: Verify that the text "Boston" always appears underlined in this page. It may also appear as strong text.

45.14.08 Class Subset 1

wcss_selectors/sel_class_subset1.xhtml

Assertion: When the Class Selector is used with an element (eg: "E.a"), then the rule must be applied to only that element matching the class attribute in the document tree.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.8.3

Verification: Verify that only the word "Boston" appears in all bold letters in this page.

45.14.09 Class Subset 2

wcss_selectors/sel_class_subset2.xhtml

Assertion: When more than one dot preceded values are used in a Class selector (eg: "E.a.b.c"), then the rule must be applied to any element whose class attribute has been assigned a list of space-separated values that include all these values in any order.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.8.3

Verification: Verify that only the word "Boston" appears in all bold letters in this page.

45.14.10 id Selector

wcss_selectors/sel_id.xhtml

Assertion: When the ID Selector is used, the rule must be applied to all elements with matching id attributes in the document tree.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.9

Verification: Verify that the only the text in "PARA-1" appears in all bold letters.

45.14.11 Grouping

wcss_selectors/sel_grouping.xhtml

Assertion: When Grouping of elements is used, the rule must be applied to all elements of the group.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.2.1

Verification: Verify that the only the word "Boston" appears always underlined in this page. Some of the words may also appear as strong or emphasized.

45.14.12 **Style Element** **wcss_selectors/style_elem.xhtml**

Assertion: When a property and its value are directly assigned to an element using STYLE attribute, then the rule must be applied to the contents of that element.

Traceability:

[WAPCSS]

Section 5.1

[CSS2]

Section 5.2.1

Verification: Verify that the only the word "Boston" appears underlined in the two latter paragraphs in this page. It may also appear as strong text.

5 SYNTAX AND BASIC DATA TYPES

5.1 Syntax and Parsing

Each test in this section refers to SCR: WCSS-SYNTAX-C-001. Start pages are described below (45.15.01 to 45.15.36). These tests include the negative tests for testing the user-agent's handling of invalid syntax.

45.15.01 Unknown Property wcss_syntax/un_property.xhtml

Assertion: When a cascading style sheet contains unknown properties, the user agent must ignore them.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.2

Verification: Verify that the word "Boston" appears in all bold letters.

45.15.02 Unknown Value wcss_syntax/un_value.xhtml

Assertion: When a cascading style sheet contains illegal values, the user agent must ignore them.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.2

Verification: Verify that the word "Boston" appears in bold letters and without any borderlines around.

45.15.03 Invalid Keyword wcss_syntax/invalid_keyword.xhtml

Assertion: When a cascading style sheet contains invalid keywords, the user agent must ignore them.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.2

Verification: Verify that the text "Boston" does not appear in bold letters.

45.15.04 Invalid at-keyword

wcss_syntax/invalid_at_key.xhtml

Assertion: When a cascading style sheet contains invalid at-keywords, the user agent must ignore it together with everything following it, up to and including the next semicolon (;) or block ({...}), whichever comes first.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.2

Verification: Verify that the text "Boston" appears underlined, but does not appear in all bold letters.

45.15.05 CSS Comments

wcss_syntax/css_comments.xhtml

Assertion: The contents of CSS comments will have no influence on the rendering of the document by the user agent.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.9

Verification: Verify that only the word "Boston" appears in bold letters.

45.15.06 Double Quotes

wcss_syntax/dquotes.xhtml

Assertion: String value of a property in a style sheet must be enclosed with either double quotes or single quotes.

Traceability:

[WAPCSS]

Section 6.1, Section 19.2

[CSS2]

Section 4.3.10

Verification: Verify that the input to the input field is limited 10 digits in the format "NNN-NNN-NNNN". Verify that the character "-" is pre-displayed at 4th and 8th positions. Submit the form to post results from next page. (This test assumes that user-agent supports "-wap-input-format" property.)

45.15.07 Single Quotes

wcss_syntax/squotes.xhtml

Assertion: String value of a property in a style sheet must be enclosed with either double quotes or single quotes.

Traceability:

[WAPCSS]

Section 6.1, Section 19.2

[CSS2]

Section 4.3.10

Verification: Verify that the input to the input field is limited 10 digits in the format “NNN-NNN-NNNN”. Verify that the character “-” is pre-displayed at 4th and 8th positions. Submit the form to post results from next page. (This test assumes that user-agent supports “-wap-input-format” property.)

45.15.08 Nested Quotes 1

wcss_syntax/nested_quotes1.xhtml

Assertion: A string value for a property can contain unescaped single quotes, if the value itself is enclosed in double quotes. (eg: A [title="my 'best' book"])

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form “E [att=val]”, then verify that “Para-1” is displayed in all bold letters.

45.15.09 Nested Quotes 2

wcss_syntax/nested_quotes2.xhtml

Assertion: A string value for a property can contain unescaped double quotes, if the value itself is enclosed in single quotes. (eg: A [title='my "best" book'])

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form “E [att=val]”, then verify that “Para-1” is displayed in all bold letters.

45.15.10 Escape Sequence 1

wcss_syntax/esc_char1.xhtml

Assertion: A double quote must be escaped using backslash character, when it is used as part of double quote enclosed string value of a property in a style sheet.
(eg: A [title="King \"Arthur\""])

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form "E [att=val]", then verify that "Para-1" is displayed in all bold letters.

45.15.11 Escape Sequence 2

wcss_syntax/esc_char2.xhtml

Assertion A single quote must be escaped using backslash character, when it is used as part of single quote enclosed string value of a property in a style sheet.
(eg: A [title='It\'s a miracle'])

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form "E [att=val]", then verify that "Para-1" is displayed in all bold letters.

45.15.12 New Line 1

wcss_syntax/new_line1.xhtml

Assertion: If the string value of a property in a style sheet has to be broken into several lines, the line breaks must be escaped using backslash character and the user-agent must treat the entire value as a string ignoring the line breaks.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form "E [att=val]", then verify that "Para-1" is displayed in bold letters.

45.15.13 New Line 2

wcss_syntax/new_line2.xhtml

Assertion: If the string value of a property in a style sheet should contain line breaks as part of its value, then the line breaks must be escaped using "\A" character and the user-agent must treat the property value including the line breaks.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.3.10

Verification: Verify that the page loads without any errors. If the user-agent supports the attribute selector of the form "E [att=val]", then verify that "Para-1" is displayed in bold letters.

45.15.14 Parsing 1

wcss_syntax/parsing1.xhtml

Assertion: When multiple properties are assigned to an element in a style sheet, and if some of them are invalid, then the user-agent must apply the remaining valid properties to the element.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.2

Verification: Verify "Para-1" is displayed in all bold letters, in default color font and with a transparent background.

45.15.15 Parsing 2

wcss_syntax/parsing2.xhtml

Assertion: When multiple values are assigned to an element in a style sheet using a shorthand property, and if some of the values are invalid, then the user-agent must apply the remaining valid values to the element.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.2

Verification: Verify the text in "Para-1" is displayed in larger and bolder font than its surrounding text. Verify that the display of text in "Para-1" is neither in italic font nor in small capitals.

45.15.16 Parsing 3

wcss_syntax/parsing3.xhtml

Assertion: When a selector name contains invalid syntax, the user-agent must ignore that selector and parse the document without breaking down.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.3

Verification: Verify that the text in "Para-1" to "Para-3" is neither displayed in red color font nor in all bold letters. Verify that only the text in "Para-4" is displayed in red colored font and in bold letters.

45.15.17 Parsing 4

wcss_syntax/parsing4.xhtml

Assertion: When a selector name contains character entities, the user-agent must interpret them with the characters they represent and must match with the matching elements.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.3

Verification: Verify that the text in "Para-1" to "Para-3" is displayed in red colored font and in bold letters.

45.15.18 SGML Comments

wcss_syntax/sgml_comments.xhtml

Assertion: When SGML comment delimiters appear in a style sheet, then user-agent must not treat them as CSS comment delimiters.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.9

Verification: Verify that each of the "Para-1" to "Para-3" is displayed in all bold text.

45.15.19 Mime Media Type

wcss_syntax/mime_type.xhtml

Assertion: The mime media type “text/css” can be used to represent WAP CSS style sheets over the Internet.

Traceability:

[WAPCSS]

Section 4.3

[CSS2]

Section 3.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters.

45.15.20 HTTP Charset Latin1

wcss_syntax/httpcharset_latin1.xhtml

Assertion: When an HTTP content-type entity-header is used to indicate the character encoding of the external CSS document, then the user-agent must use its value to determine the CSS document's character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in Latin1 encoding format).

45.15.21 HTTP Charset UTF-8

wcss_syntax/httpcharset_utf8.xhtml

Assertion: When an HTTP content-type entity-header is used to indicate the character encoding of the external CSS document, then the user-agent must use its value to determine the CSS document's character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in UTF-8 encoding format).

45.15.22 At Charset Latin1

wcss_syntax/atcharset_latin1.xhtml

Assertion 1: When @charset at-rule is used in the external CSS document to indicate its character encoding, then the user-agent must use its value to determine the document's character encoding.

Assertion 2: At most one @charset rule may appear in an external style sheet.

Assertion 3: The @charset rule must appear at the very start of the document, not preceded by any characters.

Assertion 4: After "@charset", the name of a character encoding must be specified and the name must be a charset name as described in the IANA registry.

Traceability of above 4 assertions:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in Latin1 encoding format).

45.15.23 At Charset UCS-2 BE

wcss_syntax/atcharset_ucs2_be.xhtml

Assertion: When @charset at-rule is used in the external CSS document to indicate its character encoding, then the user-agent must use its value to determine the document's character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in UCS-2 bigendian encoding format).

45.15.24 At Charset UCS-2 LE

wcss_syntax/atcharset_ucs2_le.xhtml

Assertion: When @charset at-rule is used in the external CSS document to indicate its character encoding, then the user-agent must use its value to determine the document's character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the page loads without any errors. (The external style sheet used in this test is encoded in UCS-2 littleendian encoding format). Note that para-1 may get displayed in all bold letters.

45.15.25 At Charset UTF-8

wcss_syntax/atcharset_utf8.xhtml

Assertion: When @charset at-rule is used in the external CSS document to indicate its character encoding, then the user-agent must use its value to determine the document's character encoding.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in UTF-8 encoding format).

45.15.26 **Link Charset Latin1** **wcss_syntax/linkcharset_latin1.xhtml**

Assertion: When "charset" attribute of "link" element is used to indicate the character encoding used in the external CSS document, then the user-agent must use its value to determine the linked CSS document's character encoding.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in Latin1 encoding format).

45.15.27 **Link Charset UCS-2 BE** **wcss_syntax/linkcharset_ucs2_be.xhtml**

Assertion: When "charset" attribute of "link" element is used to indicate the character encoding used in the external CSS document, then the user-agent must use its value to determine the linked CSS document's character encoding.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in UCS-2 bigendian encoding format).

45.15.28 Link Charset UCS-2 LE

wcss_syntax/linkcharset_ucs2_le.xhtml

Assertion: When “charset” attribute of “link” element is used to indicate the character encoding used in the external CSS document, then the user-agent must use its value to determine the linked CSS document’s character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the page loads without any errors. (The external style sheet used in this test is encoded in UCS-2 littleendian encoding format). Note that para-1 may get displayed in all bold letters.

45.15.29 Link Charset UTF-8

wcss_syntax/linkcharset_utf8.xhtml

Assertion: When “charset” attribute of “link” element is used to indicate the character encoding used in the external CSS document, then the user-agent must use its value to determine the linked CSS document’s character encoding.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external style sheet used in this test is encoded in UTF-8 encoding format).

45.15.30 Encoding Priority 1

wcss_syntax/enc_priority1.xhtml

Assertion: When a style sheet resides in a separate file, user agents must observe the following priorities when determining a document's character encoding (from highest priority to lowest):

- An HTTP "charset" parameter in a "Content-Type" field.
- The @charset at-rule.
- Mechanisms of the language of the referencing document (e.g., in HTML, the "charset" attribute of the LINK element).

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters.

45.15.31 Encoding Priority 2

wcss_syntax/enc_priority2.xhtml

Assertion: When a style sheet resides in a separate file, user agents must observe the following priorities when determining a document's character encoding (from highest priority to lowest):

- An HTTP "charset" parameter in a "Content-Type" field.
- The @charset at-rule.
- Mechanisms of the language of the referencing document (e.g., in HTML, the "charset" attribute of the LINK element).

Traceability:

[WAPCSS]
[CSS2]

Section 6.1
Section 4.4

Verification: Verify that the text in "Para-1" is displayed in all bold letters.

45.15.32 Multi atCharset

wcss_syntax/atcharset_multi.xhtml

Assertion: At most one @charset rule may appear in an external style sheet.

Traceability:

[WAPCSS]
[CSS2]

Section 6.1
Section 4.4

Verification:

Implementations supporting CSS 2.0 (Browsing 2.1/2.2/2.3): Verify that the page loads without errors.

Implementations supporting CSS 2.1 (Browsing 2.4): Verify that the text in "Para-1" is displayed in all bold letters (This is a negative test which contains more than one @charset at-rules in the external CSS).

45.15.33 Embedded atCharset

wcss_syntax/atcharset_embed.xhtml

Assertion: The @charset rule must not appear in an embedded style sheet.

Traceability:

[WAPCSS]
[CSS2]

Section 6.1
Section 4.4

Verification: Verify that the page loads without any errors. The text in "Para-1" may or may not get displayed in all bold letters (This is a negative test which contains @charset at-rule in the embedded style sheet).

45.15.34 atCharset After

wcss_syntax/atcharset_after.xhtml

Assertion: The @charset rule must appear at the very start of the document, not preceded by any characters..

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Verification: Verify that the page loads without any errors. The text in "Para-1" may or may not get displayed in all bold letters (This is a negative test which contains @charset at-rule preceded by other characters in the embedded style sheet).

45.15.35 Unrepresented Chars

wcss_syntax/enc_unrep_chars.xhtml

Assertion 1: When a style sheet has to refer to characters that cannot be represented in the current character encoding, then such characters must be written as escaped references to ISO 10646 characters.

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.4

Assertion 2: When a digit or letter follows the character to be escaped, then the end of the hexadecimal number needs to be made clear using either of the following two methods:

- with a space (or other whitespace character): "\26 B" ("&B")
- by providing exactly 6 hexadecimal digits: "\000026B" ("&B")

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.3

Verification: Verify that the text in "Para-1" is displayed in all bold letters. (The external CSS in this test is encoded in Latin1 format. It contains Greek characters δ and θ represented in escaped form.)

45.15.36 Invalid Escaping

wcss_syntax/enc_invalid_escape.xhtml

Assertion: When a character needs to be escaped (using backward slash followed by hexadecimal number) in a style sheet and if a digit or letter follows the character to be escaped, then the end of the hexadecimal number needs to be made clear using either of the following two methods:

- with a space (or other whitespace character): "\26 B" ("&B")
- by providing exactly 6 hexadecimal digits: "\000026B" ("&B")

Traceability:

[WAPCSS]

Section 6.1

[CSS2]

Section 4.1.3

Verification: Verify that the text in "Para-1" is not displayed in all bold letters. (This is a negative test. The external CSS document contains Greek characters δ and θ represented in escaped form. The end of the escaping hexadecimal number is intentionally left unclear.)

5.2 Data Types

All of the tests for DATA TYPES refer to SCR: WCSS-TYPES-C-001. The data types are tested in the tests of other relevant properties as follows:

- <number> as integer: Tested in –WAP-MARQUEE-LOOP property test. (test number: 45.26.06)
- <number> as real number: Tested in absolute units tests of [CSSMP] (test number: 45.32.03).
- <length> in "ex", "px" and "em" units – Tested in BORDER-xxxx- WIDTH and MARGIN-xxxx-WIDTH property tests. (# 45.19.18 and #45.19.01).
- <percentage> - tested in MARGIN-TOP-WIDTH property test. (#45.19.02).
- <uri> - tested in BACKGROUND-IMAGE and LIST-STYLE-IMAGE property tests (# 45.20.05 and #45.22.11).
- <color> as name – tested in COLOR property tests (# 45.20.02 and #45.20.03).
- <color> as numeric spec. - tested in COLOR property test (# 45.20.01).
- <string> - tested in –WAP-INPUT-FORMAT property test (#45.28.13)

Note that it is impractical to determine the exact dimensions when we test the number, length and percentage values on small devices. Instead of that, comparing the effect of two different values of same units for a property is more practical and easier to test.

For example, it is difficult for a tester to decide if the border-width is exactly rendered 5px width or 10px width. Instead, this property (border-width in this case) can be tested by comparing the display of two borders, one with 5 px and another with 10 pixels and making sure that the 10px border is displayed thicker than the 5px border.

Note that the tests described in Section 22 for the additional length units of [CSSMP] work as the negative tests for the [WAPCSS] supporting user-agent.

6 ASSIGNING PROPERTY VALUES, CASCADING AND INHERITANCE

This section contains the basic tests for testing the UA's support for the cascading and inheritance requirements. Each test in this section refers to SCR: WCSS-CASC-C-001. Start pages of individual tests are described below (45.16.01 to 45.16.11). Note that additional tests for testing the inheritance of each and every [WAPCSS] property are covered in Section 20 of this document.

45.16.01 Cascade and Inheritance `wcss_cascade/precedence.xhtml`

Assertion 1: When a user agent has parsed a document and constructed a document tree, it must assign, for every element in the tree, a value to every property that applies to the target media type by respecting the rules: 1) If the cascade results in a value, use it; 2) if the property is inherited, use it; 3) use the property's initial value.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.1.1

Assertion 2: When a property defined as inherited is set in an element, all the children elements, which do not overwrite the property, must use the same property value as defined in the parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 3: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that entire text in "PARA-1" to "PARA-3" is displayed in bold letters only and text in "Para-4" in regular case letters. Note that the text of all four paragraphs may also appear emphasized.

45.16.02 No inheritance [wcscascade/noinheritance..xhtml](#)

wcss cascade/no inheritance..xhtml

Assertion: User-agent must not assign values of non-inheritable properties of parent element to descendent elements.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.2

Verification: Verify that "Block-1" and "Block-2-Parent" are displayed with solid borders on all sides. Verify that "Block-2-Child" is contained within the borders of "Block-2-Parent" and does not have separate borders.

45.16.03 At import rule `wcss cascade/at import.xhtml`

wcss cascade/at import.xhtml

Assertion 1: When a style sheet uses the direct access of the `@import` rule, the user agent must apply the imported rules to the document.

Assertion 2: When a style sheet uses the URL access of the @import rule, the user agent must apply the imported rules to the document.

Traceability for both of above assertions is:

[WAPCSS]

Section 7

[CSS2]

Section 6.3

Verification: Verify that "Para-1" is displayed in all italic text, "PARA-2" in all bold text and "Para-3" in regular text.

45.16.04 Supported Media [wcss cascade/at import support.xhtml](#)

wcss cascade/at import support.xhtml

Assertion: When a style sheet uses the URL access of the @import rule for a specified media, the user agent must apply the imported rules to the document.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.3

Verification: Verify that the "PARA-1" is displayed in all bold letters.

45.16.05 Un-supported Media

wcss_cascade/at_import_unsupport.xhtml

Assertion: When a @import rule specifies resources from un-supported media, the user agent should not retrieve the resources.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.3

Verification: Verify that the "Para-1" is displayed just as a regular text, not in bold text. Verify that the user-agent has not made the request for CSS file "url_access.css" (You may check the gateway logs to verify this).

45.16.06 Computed values

wcss_cascade/inherit_compute.xhtml

Assertion: When the value 'inherit' is set on an element for a property, it must inherit the computed value of the property from its parent element.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.2.1

Verification: Verify that the "Word-1" is displayed in bold text. Verify that the "Word-2" and "Word-3" are displayed in normal text.

45.16.07 Cascade1

wcss_cascade/cascade1.xhtml

Assertion: When the same property is specified for an element E, in external style sheet, internal style sheet and in style attribute, then the values specified in the internal style sheet over-ride that of external style sheet and the values specified in the style attribute over-ride that of internal style sheet.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.4

Verification: Verify that the word Boston appears in red colored and bold in "Para-1". Verify that rest of text appears in all italic and green color.

45.16.08 Cascade 2

wcss_cascade/cascade2.xhtml

Assertion : The !important declaration takes precedence over a normal declaration for the value of a property.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.4.2

Verification: Verify that only the word Boston appears in red colored and all bold in "Para-1".

45.16.09 Specificity 1

wcss_cascade/speci1.xhtml

Assertion : When two rules have the same weight, origin and specificity, the later specified wins.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.4.2

Verification: Verify that the word Boston appears in red colored and bold in "Para-1".

45.16.10 Specificity 2

wcss_cascade/speci2.xhtml

Assertion : The values defined for a property in more specific selectors will over-ride those specified in less specific selectors.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.4.2

Verification: Verify that the word Boston appears in red colored and bold in "Para-1".

45.16.11 Multiple Inherit

wcss_cascade/multi_inh.xhtml

Assertion : When a property defined as inherited is set in an element, all the children elements, which do not overwrite the property, must use the same property value as defined in the parent element. This rule applies recurrently to the children of each of these child elements.

Traceability:

[WAPCSS]

Section 7

[CSS2]

Section 6.2

Verification: Verify that entire text of "Para-1" to "Para-4" is displayed in all italic characters. Note that some text of these four paragraphs may also appear as emphasized or strong text.

7 MEDIA TYPES

Each test in this section refers to SCR: WCSS-MEDIA-C-001. Start pages are described below (45.17.01).

45.17.01 Media at-rule wcss_mediatypes/media_at_rule.xhtml

Assertion 1: When a @media rule is used for the 'all' media, the user agent must apply the rules to the document.

Traceability:

[WAPCSS]	Section 8
[CSS2]	Section 7.2

Assertion 2: When a @media rule is used for the 'handheld' media, the user agent must apply the rules to the document.

Traceability:

[WAPCSS]	Section 8
----------	-----------

Assertion 3: When a @media rule is used for the un-supported media, the user agent must not apply the rules to the document.

Traceability:

[WAPCSS]	Section 8
[CSS2]	Section 7.2

Assertion 4: When a @media rule specifies a set of rules used for a comma separated list of media types and if any of these media types is applicable to the user agent, then the user agent must apply the rules to the document.

Traceability:

[WAPCSS]	Section 8
[CSS2]	Section 7.2.1

Verification: Verify that "Para-1" is not displayed in all bold text. Verify that both "PARA-2" and "PARA-3" are displayed in all bold text.

8 ASSOCIATING STYLE SHEETS WITH XML DOCUMENT

Each test in this section refers to SCR: WCSS-ASOC-C-001. Start pages are described below (45.18.01 to 45.18.09).

45.18.01 XML-stylesheet P.I. wcss_xmlpi/xmlpi.xhtml

Assertion: When an XML processing instruction is used to link an external style sheet, then the user agent MUST be able to process the 'xml-stylesheet' processing instruction, access the style sheet and apply the styles to the document.

Traceability:

[WAPCSS]	Section 9
[XMLPI]	Section 1

Verification: Verify that the entire text of "PARA-1" is displayed in bold letters only.

45.18.02 XML-stylesheet 2 wcss_xmlpi/xmlpi2.xhtml

Assertion: When a non-existent CSS file is specified in href attribute of XML processing instruction, the page should be rendered ignoring the non-existent style document.

Traceability:

[WAPCSS]	Section 9
[XMLPI]	Section 1

Verification: Verify that the page loads without any errors.

45.18.03 XML-stylesheet 3 wcss_xmlpi/xmlpi3.xhtml

Assertion: When HREF attribute is missing in the XML style sheet processing instruction, then the user-agent may generate an appropriate error in loading the document, but should not crash down.

Traceability:

[WAPCSS]	Section 9
[XMLPI]	Section 1

Verification: If the user-agent is a validating one, then verify that it displays error in loading page. For non-validating user-agents, verify that the page loads without any error. Note that the "Para-1" may or may not get displayed in all bold text.

45.18.04 XML-stylesheet 4

wcss_xmlpi/xmlpi4.xhtml

Assertion: When the MEDIA attribute is missing in the XML style sheet processing instruction, user-agent MUST load the CSS file specified in the processing instruction and apply the styles to the document.

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that the entire text of "PARA-1" is displayed in bold letters only.

45.18.05 XML-stylesheet 5

wcss_xmlpi/xmlpi5.xhtml

Assertion: When the TITLE attribute is provided in the XML style sheet processing instruction, user-agent MUST load the CSS file specified in the processing instruction and apply the styles to the document.

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that the entire text of "PARA-1" is displayed in bold letters only.

45.18.06 XML-stylesheet 6

wcss_xmlpi/xmlpi6.xhtml

Assertion: When non-applicable value is given to MEDIA attribute in the XML style sheet processing instruction, then the user-agent MUST not load the CSS file specified in the processing instruction and must render the document without any errors.

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that the text of "PARA-1" is not displayed in all bold letters.

45.18.07 XML-stylesheet 7

wcss_xmlpi/xmlpi7.xhtml

Assertion: When applicable value is given to MEDIA attribute in the XML style sheet processing instruction, then the user-agent MUST use the target CSS file to apply the styles to the document..

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that the text of "PARA-1" is displayed in all bold letters only.

45.18.08 XML-stylesheet 8

wcss_xmlpi/xmlpi8.xhtml

Assertion: When ALTERNATE attribute is set to YES in the XML style sheet processing instruction, then the user-agent MUST allow the user to select this alternate style sheet as an alternative to the default style sheet.

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that entire text in the page is displayed in lower case. If the user-agent supports choosing alternate style-sheets, then select the style sheet titled "Bold" and verify that the entire text in the page turns into all bold letters.

45.18.09 XML-stylesheet 9

wcss_xmlpi/xmlpi9.xhtml

Assertion: When HREF attribute is missing in the XML style sheet processing instruction, then the user-agent may generate an appropriate error in loading the document, but should not crash down.

Traceability:

[WAPCSS]

Section 9

[XMLPI]

Section 1

Verification: Verify that page gets loaded or a loading error gets displayed.

9 BOX MODEL

This module contains tests for margin, padding and border properties (test case numbers 45.19.01 to 45.19.43) .

9.1 Margin

The margin properties are tested in this sub-module. The start pages are described below (45.19.01 to 45.19.09). Each test in this section refers to SCR: WCSS-MARGIN-C-001. Note that Section 10.7 also contains some more tests on margins.

45.19.01 Margin Top Length

wcss_boxmodel/margin/margin_top_len.xhtml

Assertion: When the MARGIN-TOP property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the top margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

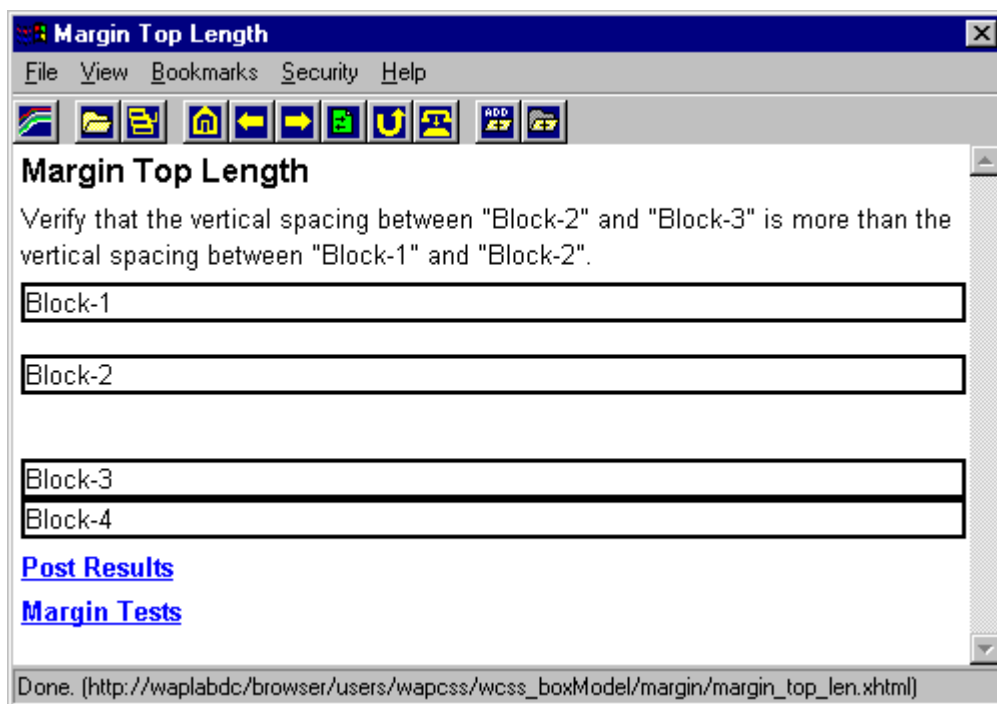
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the vertical spacing between "Block-2" and "Block-3" is much more when compared to the vertical spacing between "Block-1" and "Block-2".

Example of Display:



Assertion 1: When the, MARGIN-TOP property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the top margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 10.1

[CSS2]

Section 8.3

Assertion 2: When a property that contains a percentage value is encountered, the user agent must support that property value.

[WAPCSS]

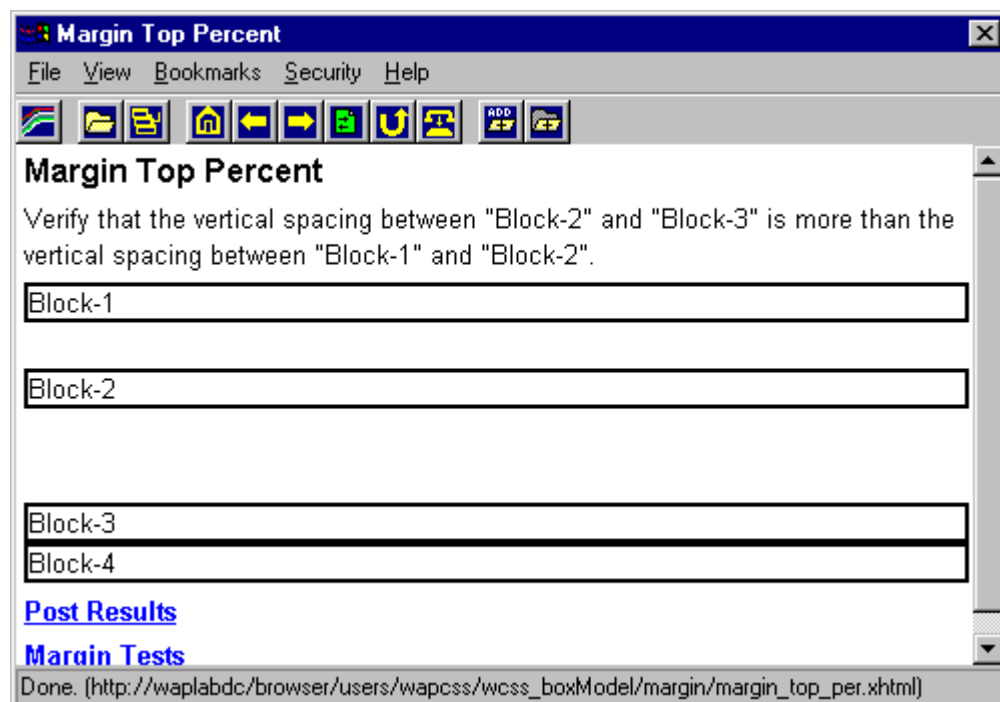
Section 6.2

[CSS2]

Section 4.3.3

Verification: Verify that the vertical spacing between "Block-2" and "Block-3" is much more when compared to the vertical spacing between "Block-1" and "Block-2".

Example of Display:



Assertion: When the MARGIN-BOTTOM property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the bottom margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

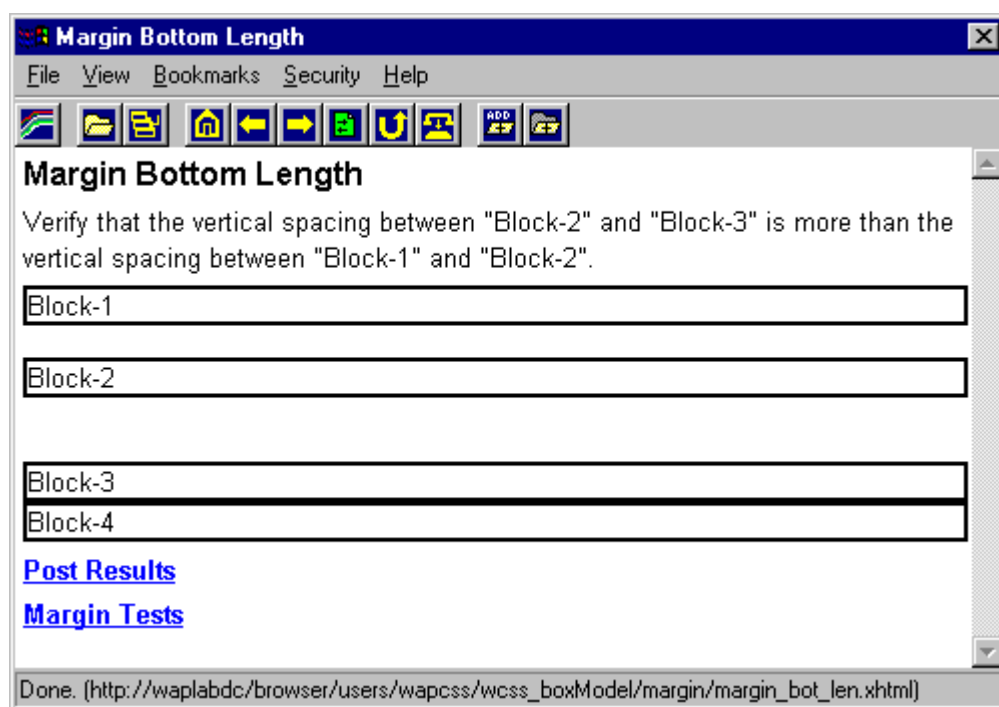
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the vertical spacing between "Block-2" and "Block-3" is much more when compared to the vertical spacing between "Block-1" and "Block-2".

Example of Display:



Assertion: When the, MARGIN-BOTTOM property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the bottom margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

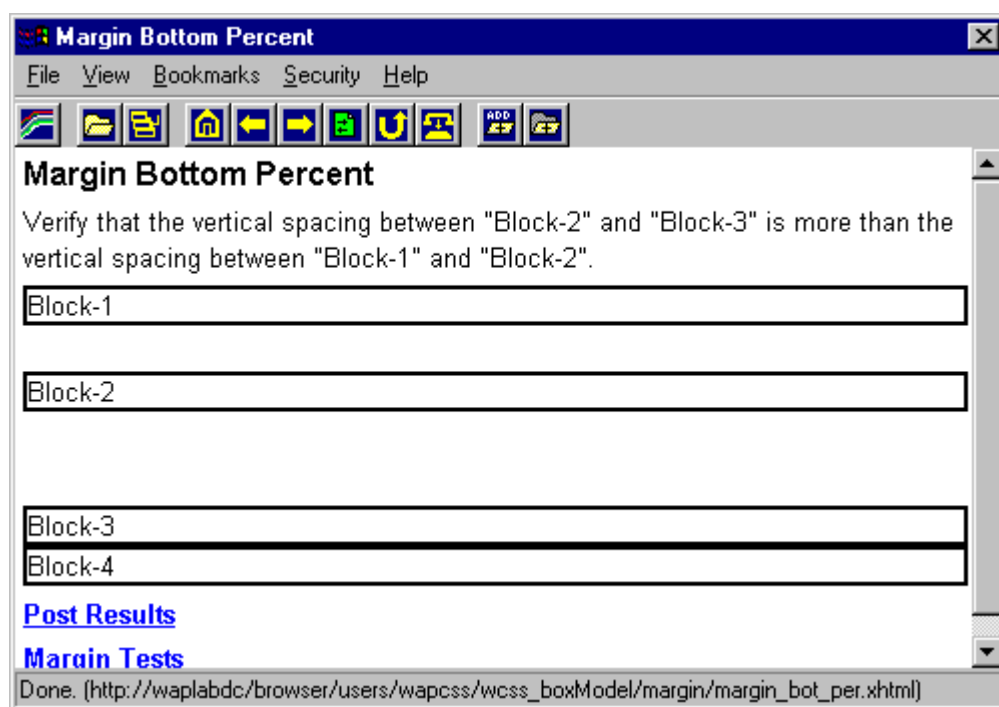
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the vertical spacing between "Block-2" and "Block-3" is much more when compared to the vertical spacing between "Block-1" and "Block-2".

Example of Display:



Assertion: When the MARGIN-LEFT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the left margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

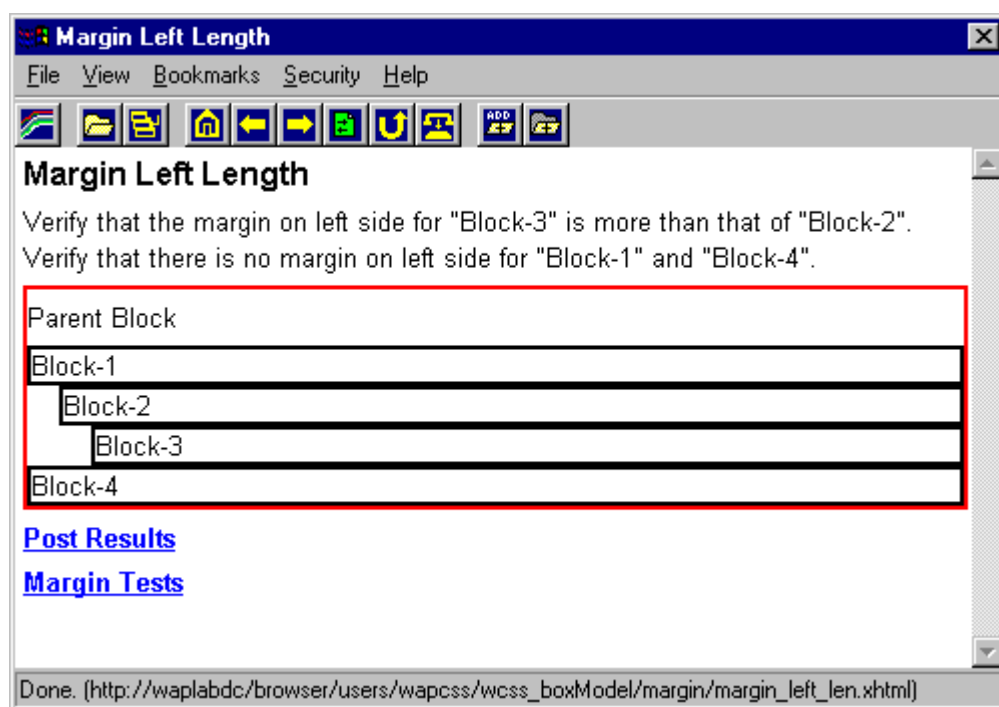
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the margin on the left side for "Block-3" is more than that of "Block-2". Verify that there is no margin on the left side for "Block-1" and "Block-4".

Example of Display:



Assertion: When the, MARGIN-LEFT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the left margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

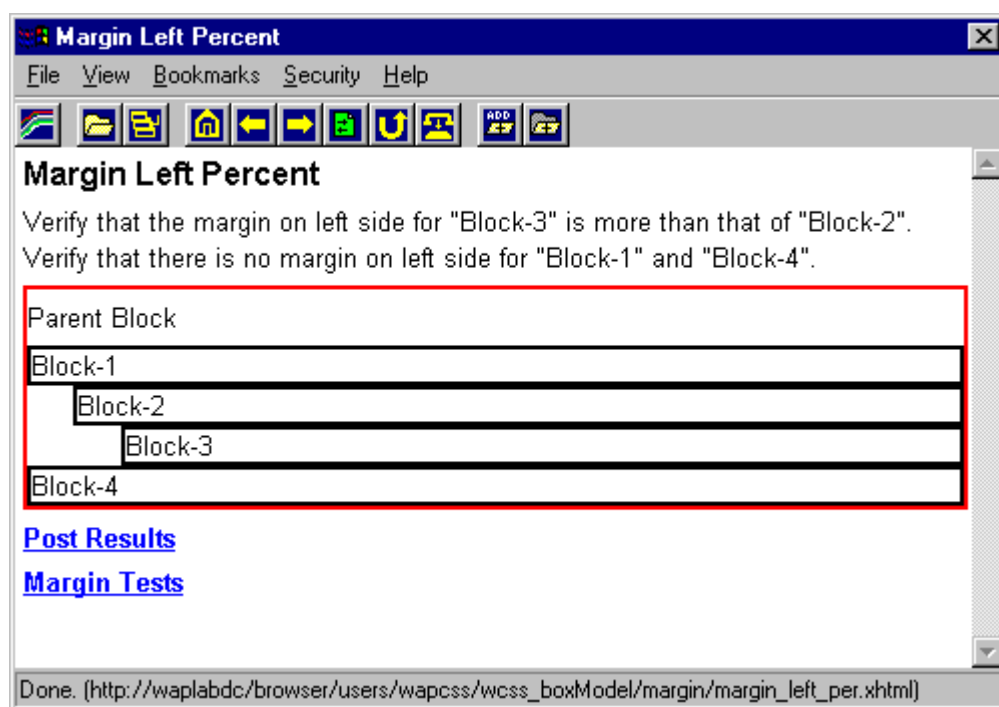
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the margin on the left side for "Block-3" is more than that of "Block-2". Verify that there is no margin on the left side for "Block-1" and "Block-4".

Example of Display:



Assertion: When the MARGIN-RIGHT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the right margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

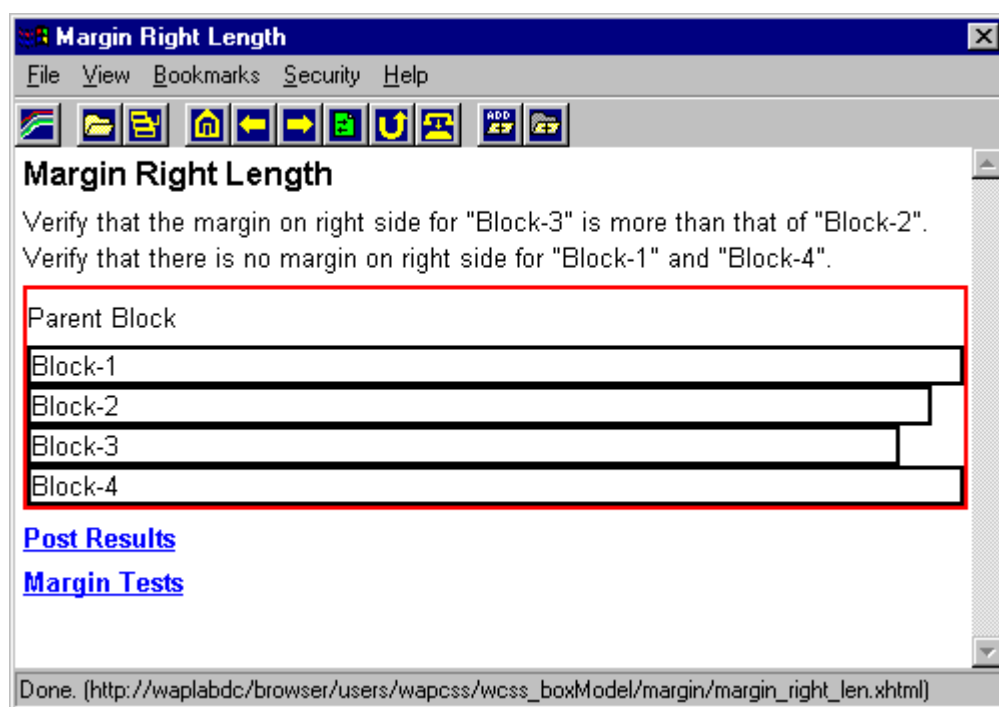
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the margin on right side for "Block-3" is more than that of "Block-2". Verify that there is no margin on right side for "Block-1" and "Block-4".

Example of Display:



Assertion: When the, MARGIN-RIGHT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the right margin) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

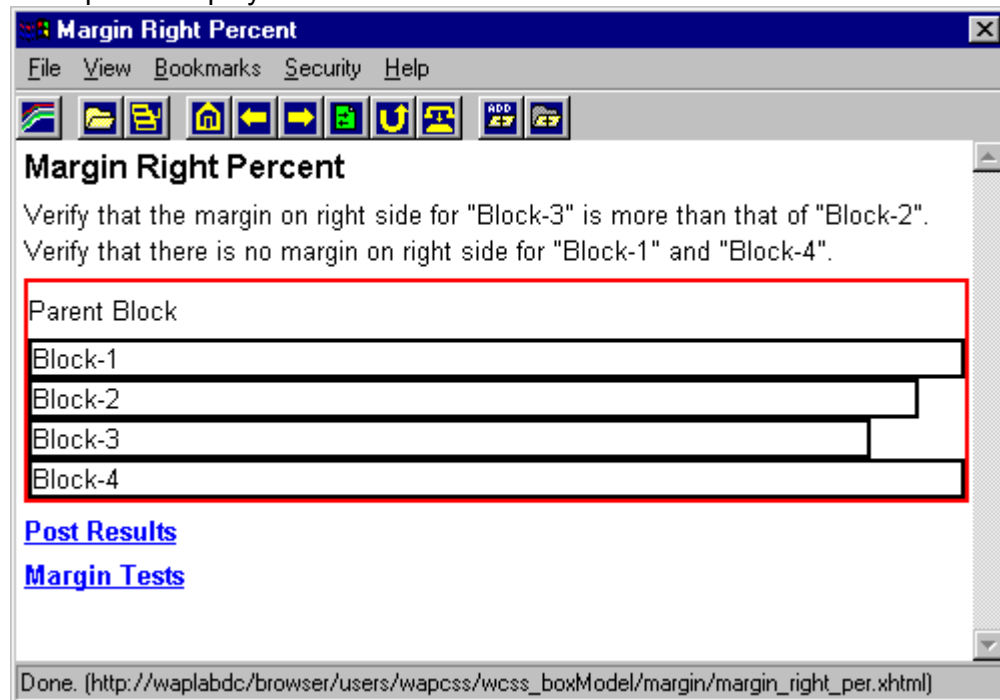
Section 10.1

[CSS2]

Section 8.3

Verification: Verify that the margin on right side for "Block-3" is more than that of "Block-2". Verify that there is no margin on right side for "Block-1" and "Block-4".

Example of Display:



Assertion 1: When the MARGIN property is set and contains one value for an element in a style sheet, the user agent must apply the value of the MARGIN property to the margin-top, margin-right, margin-left and margin-bottom for each instance of the element in the document tree.

Assertion 2: When the MARGIN property is set and contains two values for an element in a style sheet, the user agent must apply the first value of the MARGIN property to the margin-top and margin-bottom, and the second value to the margin-right and margin-left for each instance of the element in the document tree.

Assertion 3: When the MARGIN property is set and contains three values for an element in a style sheet, the user agent must apply the first value of the MARGIN property to the margin-top, the second value to the margin-right and margin-left and the third value to the margin-bottom for each instance of the element in the document tree.

Assertion 4: When the MARGIN property is set and contains four values for an element in a style sheet, the user agent must apply the first value of the MARGIN property to the margin-top, the second value to the margin-right, the third value to the margin-bottom and the fourth value to the margin-left for each instance of the element in the document tree.

Traceability for all of above 4 assertions is:

[WAPCSS]	Section 10.1
[CSS2]	Section 8.3

Verification: Verify that each of the "Block-1" to "Block-4" has margins on all four sides and they are as follows:

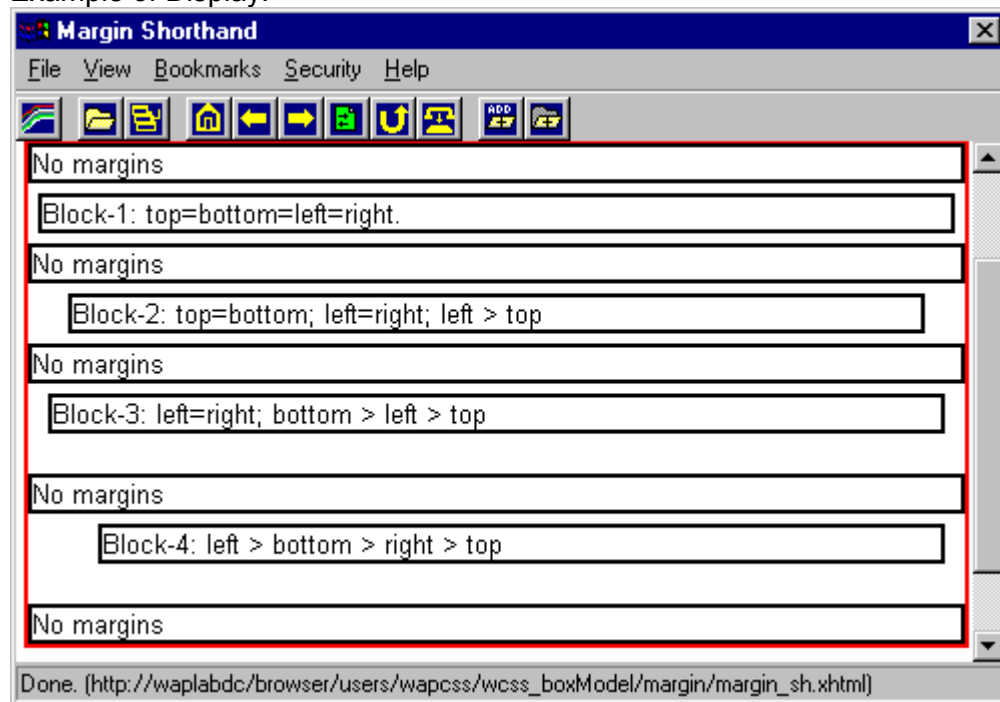
Block-1: Equal margins on all sides.

Block-2: Equal margins on top and bottom sides; equal margins on left and right sides; margins on left and right sides is more than margin on top and bottom sides.

Block-3: Equal margins on left and right sides; Bottom margin is larger than left margin and left margin is larger than top margin.

Block-4: All four margins un-equal. Left margin is the largest followed by bottom margin, right margin. Top margin is the smallest.

Example of Display:



9.2 Padding

This sub-module contains tests for padding properties. The start pages are described below (45.19.10 to 45.19.16). Each test in this section refers to SCR: WCSS-PADDING-C-001.

45.19.10 Padding Top Length

wcss_boxmodel/padding/padding_top_len.xhtml

Assertion: When the PADDING-TOP property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the top padding) and the defined value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

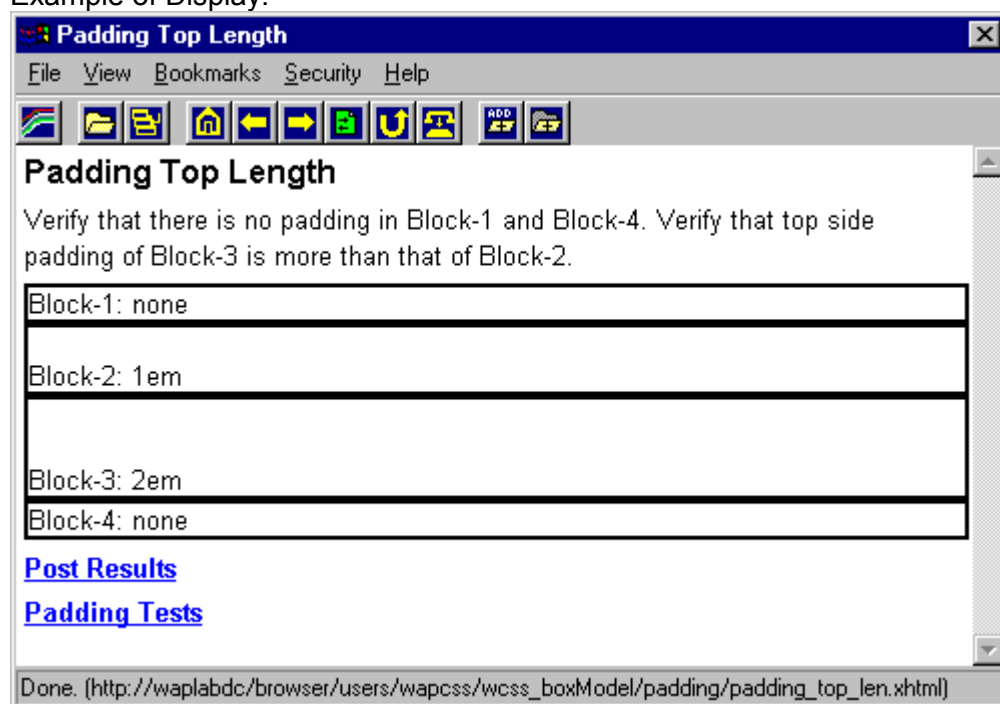
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding in "Block-1" and "Block-4" (The spacing between the border and the content). Verify that the padding on topside for "Block-3" is more than that of "Block-2".

Example of Display:



Assertion: When the PADDING-TOP property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the top padding) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

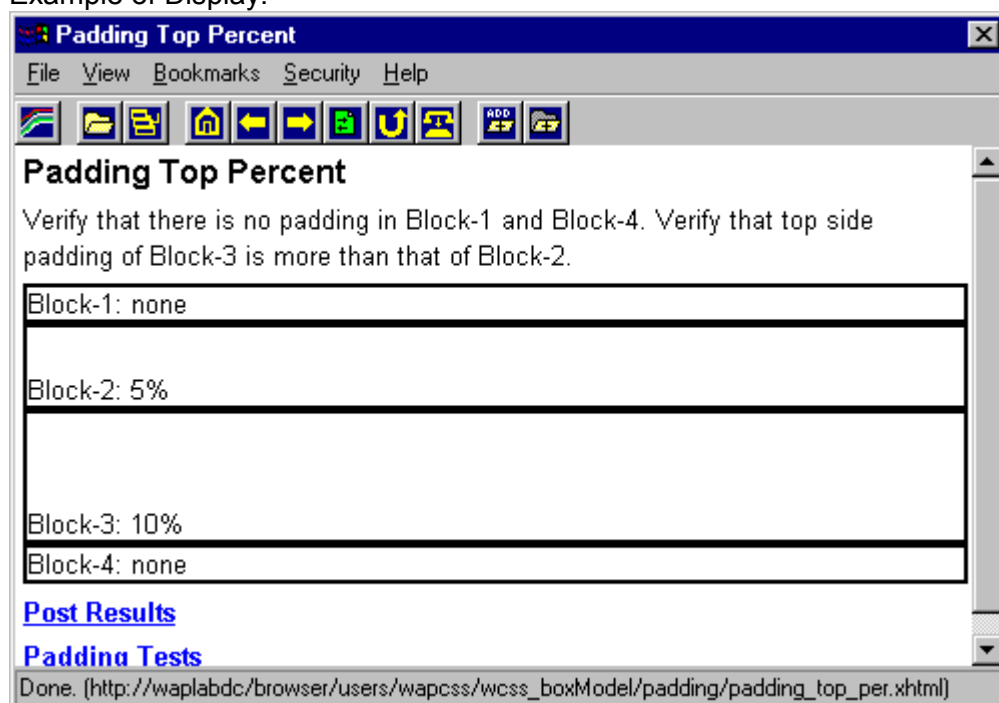
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding in "Block-1" and "Block-4" . Verify that the padding on topside for "Block-3" is more than that of "Block-2".

Example of Display:



Assertion: When the PADDING-BOTTOM property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the bottom padding) and the defined value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

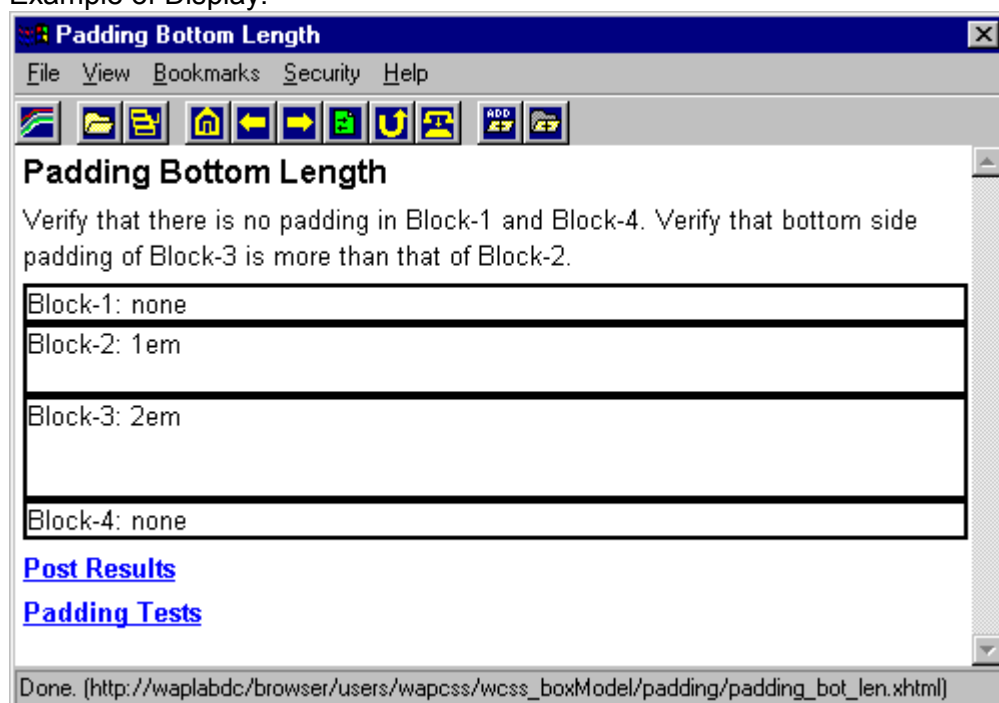
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding in "Block-1" and "Block-4". Verify that the padding on bottom side for "Block-3" is more than that of "Block-2".

Example of Display:



Assertion: When the PADDING-BOTTOM property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the bottom padding) and the computed value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

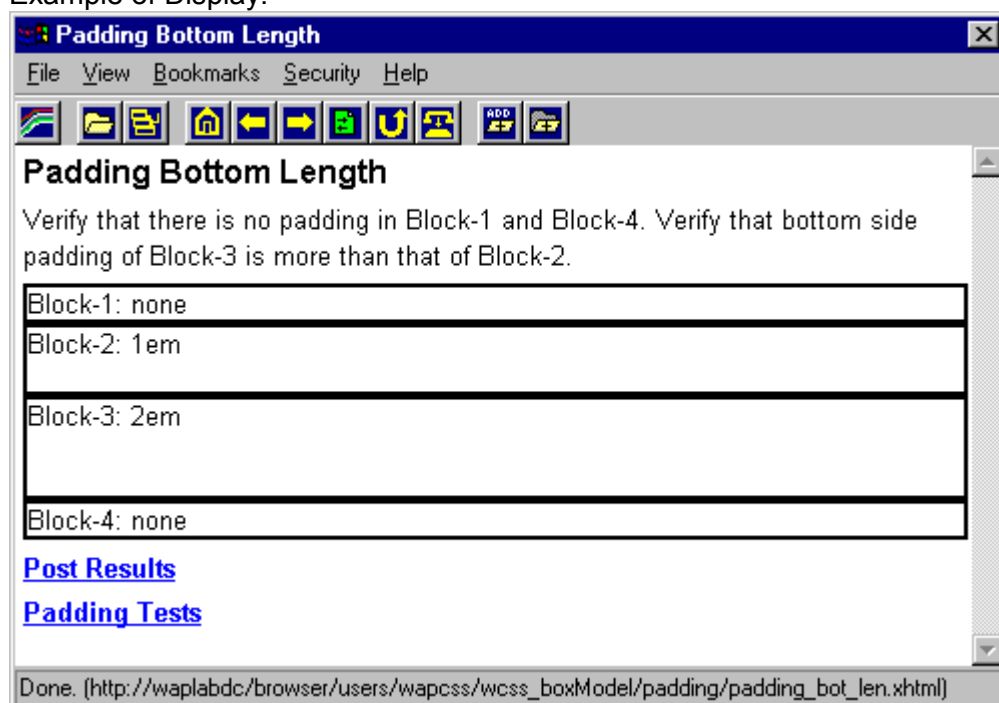
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding in "Block-1" and "Block-4". Verify that the padding on bottom side for "Block-3" is more than that of "Block-2".

Example of Display:



Assertion 1: When the PADDING-LEFT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the left padding) and the defined value for each instance of the element in the document tree.

Assertion 2: When the PADDING-LEFT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the left padding) and the computed value for each instance of the element in the document tree.

Traceability for 2 assertions is:

[WAPCSS]

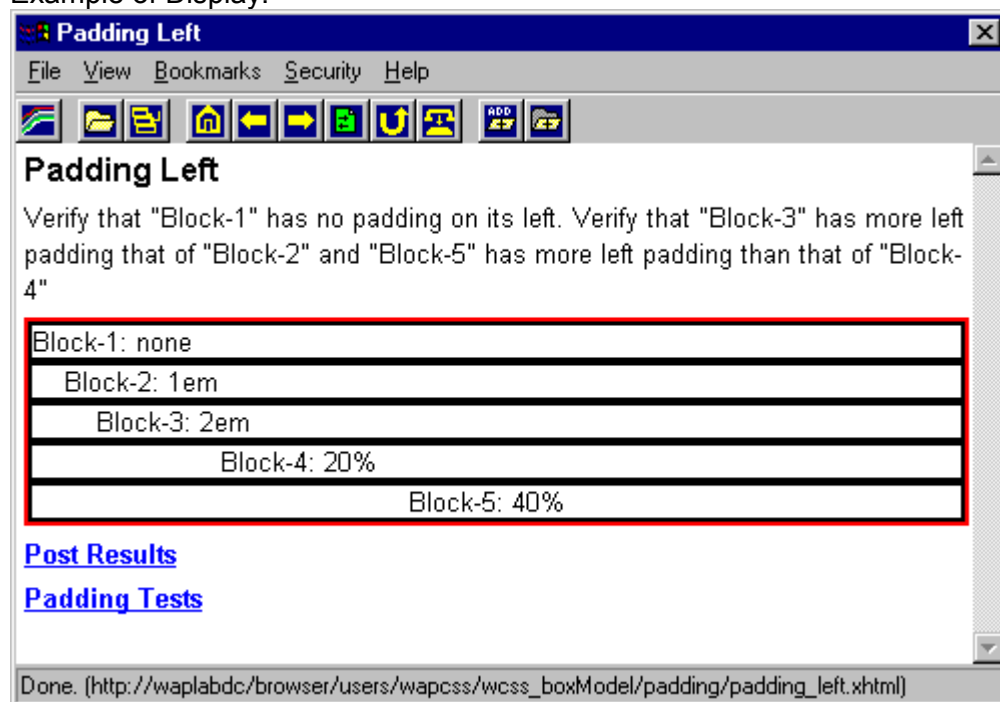
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding on left side in "Block-1". Verify that the padding on left side for "Block-3" is more than that of "Block-2". Verify that the padding on left side for "Block-5" is more than that of "Block-4".

Example of Display:



45.19.15 Padding Right

wcss_boxmodel/padding/padding_right.xhtml

Assertion 1: When the PADDING-RIGHT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the right padding) and the defined value for each instance of the element in the document tree.

Assertion 2: When the PADDING-RIGHT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the right padding) and the computed value for each instance of the element in the document tree.

Traceability of above two assertions is:

[WAPCSS]

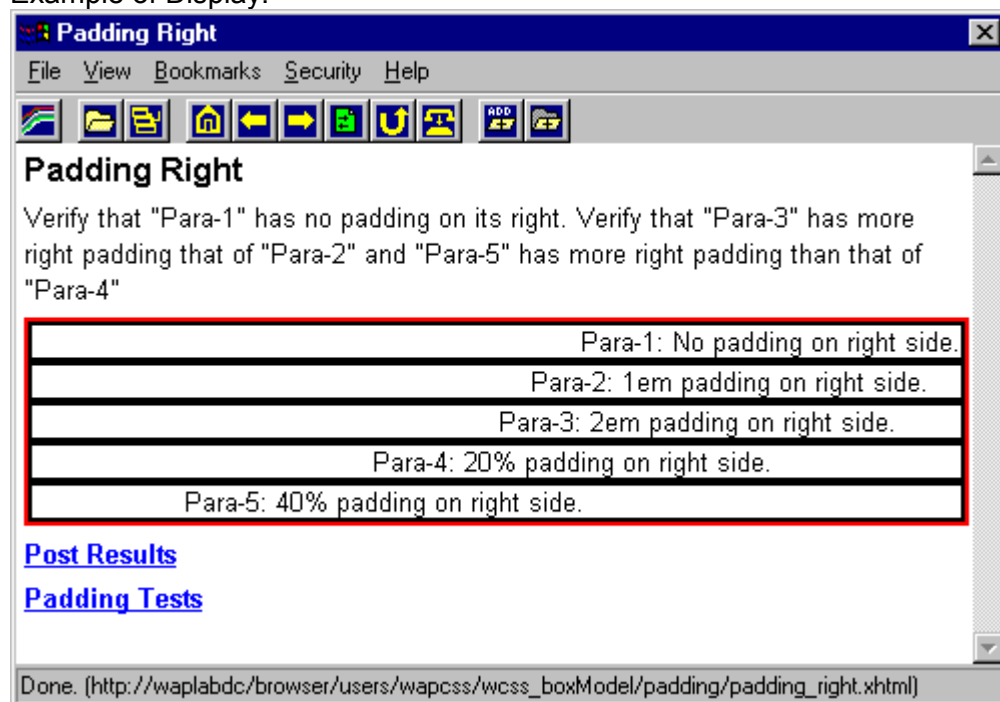
Section 10.2

[CSS2]

Section 8.4

Verification: Verify that there is no padding on right side in "Para-1". Verify that the padding on right side for "Para-3" is more than that of "Para-2". Verify that the padding on right side for "Para-5" is more than that of "Para-4".

Example of Display:



45.19.16 Padding Shorthand

wcss_boxmodel/padding/padding_sh.xhtml

Assertion 1: When the PADDING property is set and contains one value for an element in a style sheet, the user agent must apply the value of the PADDING property to the padding-top, padding-right, padding-left and padding-bottom for each instance of the element in the document tree.

Assertion 2: When the PADDING property is set and contains two values for an element in a style sheet, the user agent must apply the first value of the PADDING property to the padding-top and padding-bottom, and the second value to the padding-right and padding-left for each instance of the element in the document tree.

Assertion 3: When the PADDING property is set and contains three values for an element in a style sheet, the user agent must apply the first value of the PADDING property to the padding-top, the second value to the padding-right and padding-left and the third value to the padding-bottom for each instance of the element in the document tree.

Assertion 4: When the PADDING property is set and contains four values for an element in a style sheet, the user agent must apply the first value of the PADDING property to the padding-top, the second value to the padding-right, the third value to the padding-bottom and the fourth value to the padding-left for each instance of the element in the document tree.

Traceability for all of above 4 assertions is:

[WAPCSS]	Section 10.2
[CSS2]	Section 8.4

Verification: Verify that each of the "Block-1" to "Block-4" has two sets of borders, one outer and one inner. Verify that the padding between these two sets of borders is as described below:

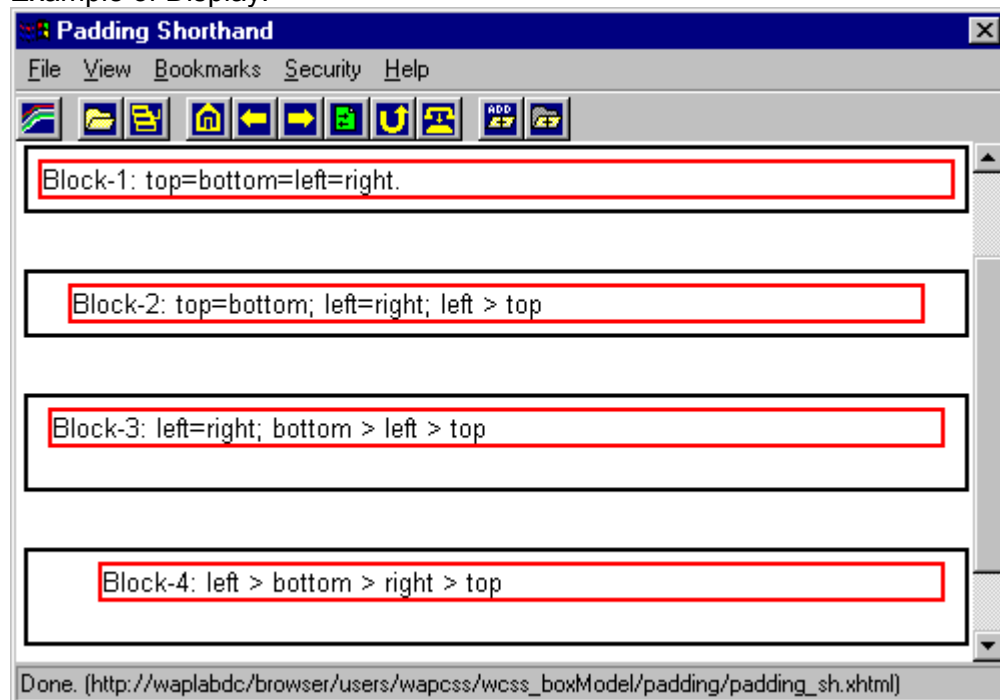
Block-1: Equal padding on all sides.

Block-2: Equal padding on top and bottom sides; equal padding on left and right sides; padding on left side is larger than padding on top side.

Block-3: Equal padding on left and right sides; Bottom padding is larger than left padding and left padding is larger than top padding.

Block-4: Padding on all four sides un-equal. Left padding is the largest followed by bottom padding, right padding. Top padding is the smallest.

Example of Display:



9.3 Border Width

This sub-module contains tests for border width properties. The start pages are described below (45.19.17 to 45.19.19). Each test in this section refers to SCR: WCSS-BORDER-WIDTH-C-001.

45.19.17 Border Width – Keywords wcss_boxmodel/border_width/bwidth_key.xhtml

Assertion 1: When the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties is set to 'thin' for an element in a style sheet, the user agent must display a 'thin' border for each instance of the element in the document tree.

Assertion 2: When the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties is set to 'medium' for an element in a style sheet, the user agent must display a 'medium' border for each instance of the element in the document tree.

Assertion 3: When the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties is set to 'thick' for an element in a style sheet, the user agent must display a 'thick' border for each instance of the element in the document tree.

Assertion 4: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'solid' for an element set in a style sheet and the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties are not, the user agent must display a 'medium' border for each instance of the element in the document tree.

Assertion 5: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'none' for an element set in a style sheet and the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties are set, the user agent must not display any border for each instance of the element in the document tree.

Traceability for above 5 assertions is:

[WAPCSS]
[CSS2]

Section 10.3
Section 8.5.1

Verification : Verify that the "Block-1" has no borders displayed on any of its sides. Verify that each of the "Block-2" to "Block-5" has single line border of equal thickness displayed on all four sides. Verify that the borders of "Block-3" and "Block-4" are of same thickness. Verify that "Block-2" has thinner borders and "Block-5" has thicker borders when compared to "Block-3".

45.19.18 **Border Width – Length**
wcss_boxmodel/border_width/bwidth_length.xhtml

Assertion 1: When a property that contains a 'px' length value is encountered, the user agent must support that property value.

Assertion 2: When a property that contains an 'ex' length value is encountered, the user agent must support that property value.

Assertion 3: When a property that contains an 'em' length value is encountered, the user agent must support that property value.

Traceability of above 3 assertions:

[WAPCSS]	Section 6.2
[CSS2]	Section 4.3.2

Assertion 4: When the BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH or BORDER-LEFT-WIDTH properties is set as a <length> value for an element in a style sheet, the user agent must display a border with the defined value for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 10.5
[CSS2]	Section 8.5.1

Verification: Verify that each of the "Block-1" to "Block-6" have borders displayed on all four sides. Verify that the borders of "Block-2" are thicker than that of "Block-1". Verify that the borders of "Block-4" are thicker than that of "Block-3". Verify that the borders of "Block-6" are thicker than that of "Block-5".

Assertion 1: When the BORDER-WIDTH property is set and contains one value for an element in a style sheet, the user agent must apply the value of the BORDER-WIDTH property to the border-top-width, border-right-width, border-left-width and border-bottom-width for each instance of the element in the document tree.

Assertion 2: When the BORDER-WIDTH property is set and contains two values for an element in a style sheet, the user agent must apply the first value of the BORDER-WIDTH property to the border-top-width and border-bottom-width, and the second value to the border-right-width and border-left-width for each instance of the element in the document tree.

Assertion 3: When the BORDER-WIDTH property is set and contains three values for an element in a style sheet, the user agent must apply the first value of the BORDER-WIDTH property to the border-top-width, the second value to the border-right-width and border-left-width and the third value to the border-bottom-width for each instance of the element in the document tree.

Assertion 4: When the BORDER-WIDTH property is set and contains four values for an element in a style sheet, the user agent must apply the first value of the BORDER-WIDTH property to the border-top-width, the second value to the border-right-width, the third value to the border-bottom-width and the fourth value to the border-left-width for each instance of the element in the document tree.

Traceability for above 4 assertions is:

[WAPCSS]

Section 10.3

[CSS2]

Section 8.5.1

Verification: Verify that border widths of the "Block-1" to "Block-4" is displayed as follows:

Block-1: Equal border width on all sides.

Block-2: Equal border width on top and bottom sides; equal border width on left and right sides; border width on left side is more than border width on top side.

Block-3: Equal border width on left and right sides; Bottom border width is larger than left border width and left border width is larger than top border width.

Block-4: Border width on all four sides un-equal. Left border width is the largest followed by bottom border width, right border width. Top border width is the smallest.

9.4 Border Color

This sub-module contains tests for border color properties. The start pages are described below (45.19.20 to 45.19.21). Each test in this section refers to SCR: WCSS-BORDER-COLOR-C-001.

45.19.20 Border Colors `wcss_boxmodel/border_color/bcolors_sides.xhtml`

Assertion 1: When the BORDER-TOP-COLOR property is set for an element in a style sheet, the user agent must apply the border-top-color rule (Apply the color value to the border-top) for each instance of the element in the document tree.

Assertion 2: When the BORDER-RIGHT-COLOR property is set for an element in a style sheet, the user agent must apply the border-right-color rule (Apply the color value to the border-right) for each instance of the element in the document tree.

Assertion 3: When the BORDER-BOTTOM-COLOR property is set for an element in a style sheet, the user agent must apply the border-bottom-color rule (Apply the color value to the border-bottom) for each instance of the element in the document tree.

Assertion 4: When the BORDER-LEFT-COLOR property is set for an element in a style sheet, the user agent must apply the border-left-color rule (Apply the color value to the border-left) for each instance of the element in the document tree.

Assertion 5: When the BORDER-TOP-COLOR, BORDER-RIGHT-COLOR, BORDER-BOTTOM-COLOR or BORDER-LEFT-COLOR properties is set as a <color> value for an element in a style sheet, the user agent must display a border with the defined color value for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 10.4
Section 8.5.2

Verification: Verify that each of the "Block-1" to "Block-5" is displayed with red colored borderlines on all four sides.

Assertion 1: When the BORDER-COLOR property is set and contains one value for an element in a style sheet, the user agent must apply the value of the BORDER-COLOR property to the border-top-color, border-right-color, border-left-color and border-bottom-color for each instance of the element in the document tree.

Assertion 2: When the BORDER-COLOR property is set and contains two values for an element in a style sheet, the user agent must apply the first value of the BORDER-COLOR property to the border-top-color and border-bottom-color, and the second value to the border-right-color and border-left-color for each instance of the element in the document tree.

Assertion 3: When the BORDER-COLOR property is set and contains three values for an element in a style sheet, the user agent must apply the first value of the BORDER-COLOR property to the border-top-color, the second value to the border-right-color and border-left-color and the third value to the border-bottom-color for each instance of the element in the document tree.

Assertion 4: When the BORDER-COLOR property is set and contains four values for an element in a style sheet, the user agent must apply the first value of the BORDER-COLOR property to the border-top-color, the second value to the border-right-color, the third value to the border-bottom-color and the fourth value to the border-left-color for each instance of the element in the document tree.

Traceability for above 4 assertions is:

[WAPCSS]

Section 10.4

[CSS2]

Section 8.5.2

Verification: Verify that the color of borderlines of "Block-1" to "Block-4" is as follows:

Block 1: red color border on all sides.

Block 2: top and bottom – red color border; left and right – blue color border.

Block 3: top – red color border; left and right – blue color border; bottom – green color border.

Block 4: top – red color border; right – blue color border; bottom – green color border; left – yellow color border.

9.5 Border Style

This sub-module contains tests for border style properties. The start pages are described below (45.19.22 to 45.19.28). Each test in this section refers to SCR: WCSS-BORDER-STYLE-C-001.

45.19.22 Border Style – Solid wcss_boxmodel/border_style/bstyle_solid.xhtml

Assertion 1: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'solid' for an element in a style sheet, the user agent must display borders as a single line element for each instance of the element in the document tree.

Assertion 2: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'none' for an element in a style sheet, the user agent must not display borders for each instance of the element in the document tree.

Traceability for above 2 assertions is:

[WAPCSS]	Section 10.5
[CSS2]	Section 8.5.3

Verification: Verify that the "Block-1" has no borders displayed on any of its sides. Verify that "Block-2" has single line solid borders displayed on all four sides.

45.19.23 Border Style – Hidden wcss_boxmodel/border_style/bstyle_hidden.xhtml

Assertion: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'hidden' for an element in a style sheet, the user agent must apply the border conflict resolution rule for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 10.5
[CSS2]	Section 8.5.3

Verification: Verify that the "Block-1" has no borders displayed on any of its sides. Verify that "Block-2" has single line solid borders displayed on all four sides.

45.19.24 **Border Style – Others 1**
wcss_boxmodel/border_style/bstyle_others1.xhtml

Assertion 1: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'dotted' for an element in a style sheet, the user agent must display borders as a series of dots for each instance of the element in the document tree.

Assertion 2: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'dashed' for an element in a style sheet, the user agent must display borders as a series of short line segments for each instance of the element in the document tree.

Assertion 3: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'double' for an element in a style sheet, the user agent must display borders as two solid lines for each instance of the element in the document tree.

Traceability for all of above 3 assertions is:

[WAPCSS]	Section 10.5
[CSS2]	Section 8.5.3

Verification: Verify that each of the "Block-1" to "Block-3" has border styles on four sides as described below. *Note that the user-agent may render all these styles as solid lines.*

Block-1: Dotted lines - series of dots.

Block-2: Dashed lines – series of short line segments.

Block-3: Double lines – two parallel solid lines.

45.19.25 **Border Style – Others 2**
wcss_boxmodel/border_style/bstyle_others2.xhtml

Assertion 1: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'groove' for an element in a style sheet, the user agent must display borders looking as though it were carved into the canvas for each instance of the element in the document tree.

Assertion 2: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'ridge' for an element in a style sheet, the user agent must display borders looking as though it were coming out of the canvas for each instance of the element in the document tree.

Assertion 3: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'inset' for an element in a style sheet, the user agent must display borders, that make the entire box look as though it were embedded in the canvas for each instance of the element in the document tree.

Assertion 4: When the BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE or BORDER-LEFT-STYLE properties is set to 'outset' for an element in a style sheet, the user agent must display borders, which make the entire box look as though it were coming out of the canvas for each instance of the element in the document tree.

Traceability for all of above 4 assertions is:

[WAPCSS]	Section 10.5
[CSS2]	Section 8.5.3

Verification: Verify that each of the "Block-1" to "Block-4" has border styles on four sides as described below. Note that the user-agent may render all these styles as solid lines.

Block-1: Grooved lines - as if carved into canvas.

Block-2: Ridge lines – as if coming out of canvas (opposite of Grooved lines).

Block-3: Inset lines - looks as if entire box is embedded in the canvas.

Block-4: Outset lines – looks as if the entire box is coming out of canvas.

Assertion 1: When the BORDER-STYLE property is set and contains one value for an element in a style sheet, the user agent must apply the value of the BORDER-STYLE property to the border-top-style, border-right-style, border-left-style and border-bottom-style for each instance of the element in the document tree.

Assertion 2: When the BORDER-STYLE property is set to 'solid' for an element in a style sheet, the user agent must display borders as a single line element for each instance of the element in the document tree.

Assertion 3: When the BORDER-STYLE property is set to 'none' for an element in a style sheet, the user agent must not display any borders for each instance of the element in the document tree.

Assertion 4: When the BORDER-STYLE property is set to 'hidden' for an element in a style sheet, the user agent must apply the border conflict resolution rule for each instance of the element in the document tree.

Traceability for above 4 assertions is:

[WAPCSS]

Section 10.5

[CSS2]

Section 8.5.3

Verification: Verify that "Block-1" is displayed with single line solid border on all sides. Verify that "Block-2" and "Block-3" has no borders displayed.

Assertion 1: When the BORDER-STYLE property is set to 'dotted' for an element in a style sheet, the user agent must display borders as a series of dots for each instance of the element in the document tree.

Assertion 2: When the BORDER-STYLE property is set to 'dashed' for an element in a style sheet, the user agent must display borders as a series of short lines segments for each instance of the element in the document tree.

Assertion 3: When the BORDER-STYLE property is set to 'double' for an element in a style sheet, the user agent must display borders as two solid lines for each instance of the element in the document tree.

Assertion 4: When the BORDER-STYLE property is set to 'groove' for an element in a style sheet, the user agent must display a border looking as though it were carved into the canvas for each instance of the element in the document tree.

Assertion 5: When the BORDER-STYLE property is set to 'ridge' for an element in a style sheet, the user agent must display a border looking as though it were coming out of the canvas for each instance of the element in the document tree.

Assertion 6: When the BORDER-STYLE property is set to 'inset' for an element in a style sheet, the user agent must display a border, that make the entire box look as though it were embedded in the canvas for each instance of the element in the document tree.

Assertion 7: When the BORDER-STYLE property is set to 'outset' for an element in a style sheet, the user agent must display a border, that make the entire box look as though it were coming out of the canvas for each instance of the element in the document tree.

Traceability for above 7 assertions is:

[WAPCSS]
[CSS2]

Section 10.5
Section 8.5.3

Verification: Verify that each of the "Block-1" to "Block-7" has border styles on four sides as described. Note that the user-agent may render all these styles as solid lines.

Block-1: Dotted lines - series of dots.

Block-2: Dashed lines – series of short line segments.

Block-3: Double lines – two parallel solid lines.

Block-4: Grooved lines - as if carved into canvas.

Block-5: Ridge lines – as if coming out of canvas (opposite of Grooved lines).

Block-6: Inset lines - looks as if entire box is embedded in the canvas.

Block-7: Outset lines – looks as if the entire box is coming out of canvas.

Assertion 1: When the BORDER-STYLE property is set and contains one value for an element in a style sheet, the user agent must apply the value of the BORDER-STYLE property to the border-top-style, border-right-style, border-left-style and border-bottom-style for each instance of the element in the document tree.

Assertion 2: When the BORDER-STYLE property is set and contains two values for an element in a style sheet, the user agent must apply the first value of the BORDER-STYLE property to the border-top-style and border-bottom-style, and the second value to the border-right-style and border-left-style for each instance of the element in the document tree.

Assertion 3: When the BORDER-STYLE property is set and contains three values for an element in a style sheet, the user agent must apply the first value of the BORDER-STYLE property to the border-top-style, the second value to the border-right-style and border-left-style and the third value to the border-bottom-style for each instance of the element in the document tree.

Assertion 4: When the BORDER-STYLE property is set and contains four values for an element in a style sheet, the user agent must apply the first value of the BORDER-STYLE property to the border-top-style, the second value to the border-right-style, the third value to the border-bottom-style and the fourth value to the border-left-style for each instance of the element in the document tree.

Traceability for above 4 assertions is:

[WAPCSS]
[CSS2]

Section 10.5
Section 8.5.3

Verification: Verify that each of the "Block-1" to "Block-4" has border styles on each side as described below. Note that browser may render all styles as solid lines except for none and hidden styles.

Block-1: solid lines all sides.

Block-2: top and bottom dotted line; left and right dashed line.

Block-3: top double; left and right groove line; bottom ridge.

Block-4: top solid; right double; bottom hidden; left none.

9.6 Border Shorthand Properties

This sub-module contains tests for short hand property 'border'. The start pages are described below (45.19.29 to 45.19.36). Each test in this section refers to SCR: WCSS-BORDER-SHORT-C-001.

45.19.29 Border-side Shorthand 1 `wcss_boxmodel/border_shand/bshort_side1.xhtml`

Assertion 1: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border with the defined style value and color value as defined in the COLOR property for each instance of the element in the document tree.

Assertion 2: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a width value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a border with the defined style value, the defined size value and whom the color is the same as defined in the COLOR property for each instance of the element in the document tree.

Assertion 3: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Assertion 4: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a color value, a width value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a border with the defined style value, the defined color value and the defined width value for each instance of the element in the document tree.

Traceability of above 4 assertions:

[WAPCSS]	Section 10.6
[CSS2]	Section 8.5.4

Verification: Verify that the each of the "Block-1" to "Block-4" has green colored text as content. Verify that they have borders displayed on all four sides and in the following style:

Block 1: Green color and medium thickness borders.
Block 2: Green color and thicker borders.
Block 3: Red color and medium thickness borders.
Block 4: Red color and thicker borders.

45.19.30 **Border-side Shorthand 2**
wcss_boxmodel/border_shand/bshort_side2.xhtml

Assertion 1: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a width value for an element in a style sheet, the user agent must display no borders for each instance of the element in the document tree as the initial style value is 'none'.

Assertion 2: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each of one, with a color value for an element in a style sheet, the user agent must display no borders for each instance of the element in the document tree as the initial style value is 'none'.

Assertion 3: When the BORDER-TOP, the BORDER-RIGHT, the BORDER-BOTTOM and the BORDER-LEFT properties are set, each one with a width and a color value for an element in a style sheet, the user agent must display no borders for each instance of the element in the document tree as the initial style value is 'none'.

Traceability of above 3 assertions:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Verification: Verify that none of the "Block-1", "Block-2" or "Block-3" is displayed with any borders.

Assertion 1: When the BORDER property is set with a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border with the defined style value and whom the color is the same as defined in the COLOR property for each instance of the element in the document tree.

Assertion 2: When the BORDER property is set with a width value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a border with the defined style value, the defined size value and whom the color is the same as defined in the COLOR property for each instance of the element in the document tree.

Assertion 3: When the BORDER property is set with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Assertion 4: When the BORDER property is set with a color value, a width value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a border with the defined style value, the defined color value and the defined width value for each instance of the element in the document tree.

Traceability of above 4 assertions:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Verification: Verify that the each of the "Block-1" to "Block-4" has green colored text as content. Verify that they have borders displayed on all four sides and in the following style:

Block 1: Green color and medium thickness borders.

Block 2: Green color and thicker borders.

Block 3: Red color and medium thickness borders.

Block 4: Red color and thicker borders.

45.19.32 Border Shorthand 2

wcss_boxmodel/border_shand/bshort_all2.xhtml

Assertion 1: When the BORDER property is set with a width value for an element in a style sheet, the user agent must display no border for each instance of the element in the document tree as the initial style value is 'none'.

Assertion 2: When the BORDER is set with a color value for an element in a style sheet, the user agent must display no borders for each instance of the element in the document tree as the initial style value is 'none'.

Assertion 3: When the BORDER property is set with a width and a color value for an element in a style sheet, the user agent must display no border for each instance of the element in the document tree as the initial style value is 'none'.

Traceability of above 3 assertions:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Verification: Verify that none of the "Block-1", "Block-2" or "Block-3" is displayed with any borders.

45.19.33 Named Colors 1

wcss_boxmodel/border_shand/col_named1.xhtml

Assertion 1: When the BORDER property is set with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that colors of the borders for the "Block-1" to "Block-5" are as described in the test.

45.19.34 Named Colors 2

wcss_boxmodel/border_shand/col_named2.xhtml

Assertion 1: When the BORDER property is set with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that colors of the borders for the "Block-1" to "Block-5" are as described in the test.

45.19.35 Named Colors 3

wcss_boxmodel/border_shand/col_named3.xhtml

Assertion 1: When the BORDER property is set with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that colors of the borders for the "Block-1" to "Block-4" are as described in the test.

Assertion 1: When the BORDER property is set with a color value and a style value different from 'none' and 'hidden' for an element in a style sheet, the user agent must display a medium size border as a single line segment with the defined color value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 10.6

[CSS2]

Section 8.5.4

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that colors of the borders for the "Block-1" to "Block-3" are as described in the test.

9.7 More Box Model Tests:

45.19.37 Collapsing Margins

wcss_boxmodel/coll_margins.xhtml

Assertion 1: When the margins are assigned to two or more adjoining boxes in the normal flow, then the user-agent must collapse the vertical margin between the adjoining boxes by taking the maximum of adjoining margins as the resulting combined margin.

Assertion 2: When the margins are assigned to two or more adjoining boxes in the normal flow, then the user-agent must not collapse the horizontal margins between the adjoining boxes.

Traceability of above two assertions:

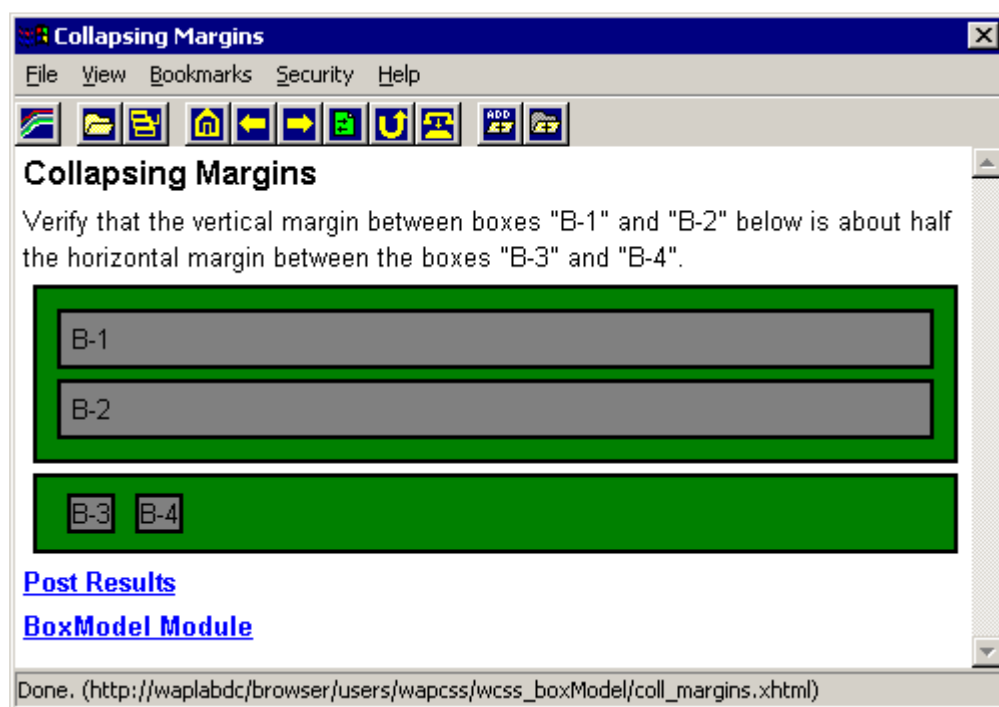
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1

Verification: Verify that the vertical margin between boxes "B-1" and "B-2" is about half the horizontal margin between the boxes "B-3" and "B-4".



Assertion: When the margins are assigned to two or more adjoining boxes in the normal flow, then the user-agent must collapse the vertical margin between the adjoining boxes by taking the maximum of adjoining margins as the resulting combined margin.

Traceability of above two assertions:

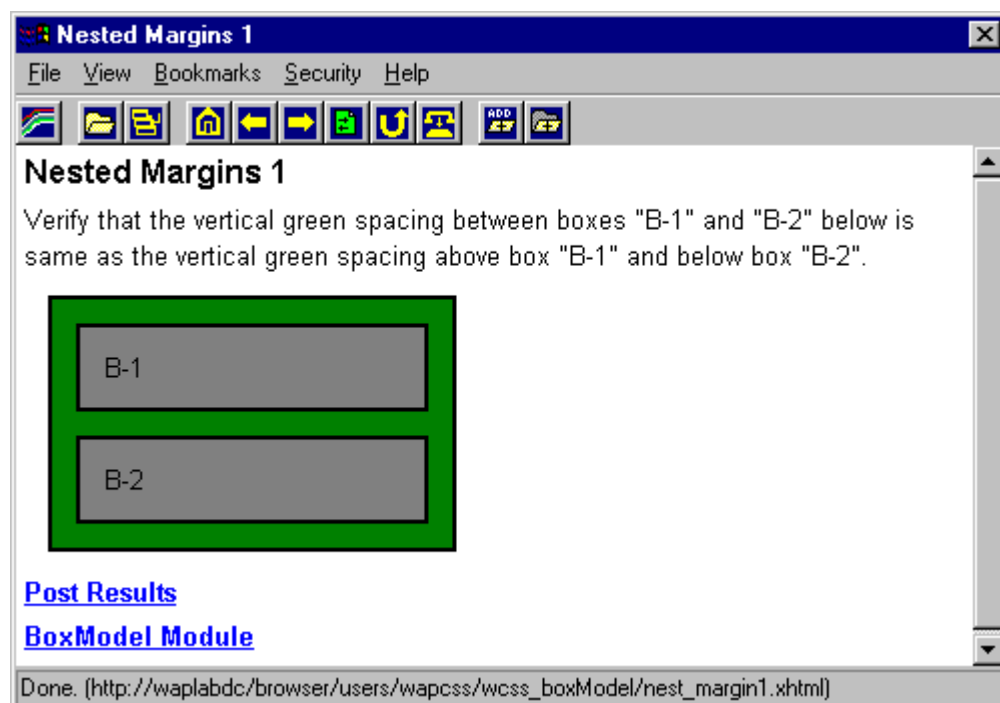
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1

Verification: Verify that the vertical green spacing between boxes "B-1" and "B-2" is same as the vertical green spacing above box B-1 and below box B-2.



Assertion: When the margins are assigned to two or more adjoining boxes in the normal flow, then the user-agent must collapse the vertical margin between the adjoining boxes by taking the maximum of adjoining margins as the resulting combined margin.

Traceability of above two assertions:

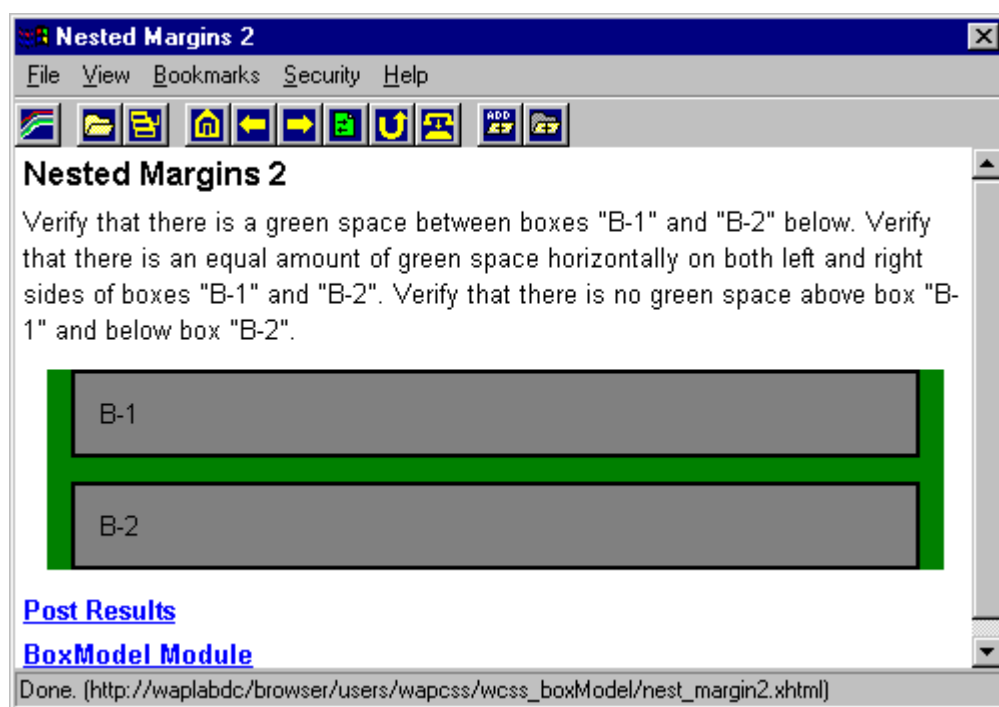
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1

Verification: Verify that there is a green space between boxes "B-1" and "B-2". Verify that there is an equal amount of green space horizontally on both left and right sides of boxes "B-1" and "B-2". Verify that there is no green space above box "B-1" and below box "B-2".



Assertion : When the margins are assigned to floating boxes and its containing box, then the user-agent must not collapse the vertical margins between the float box and its containing box.

Traceability:

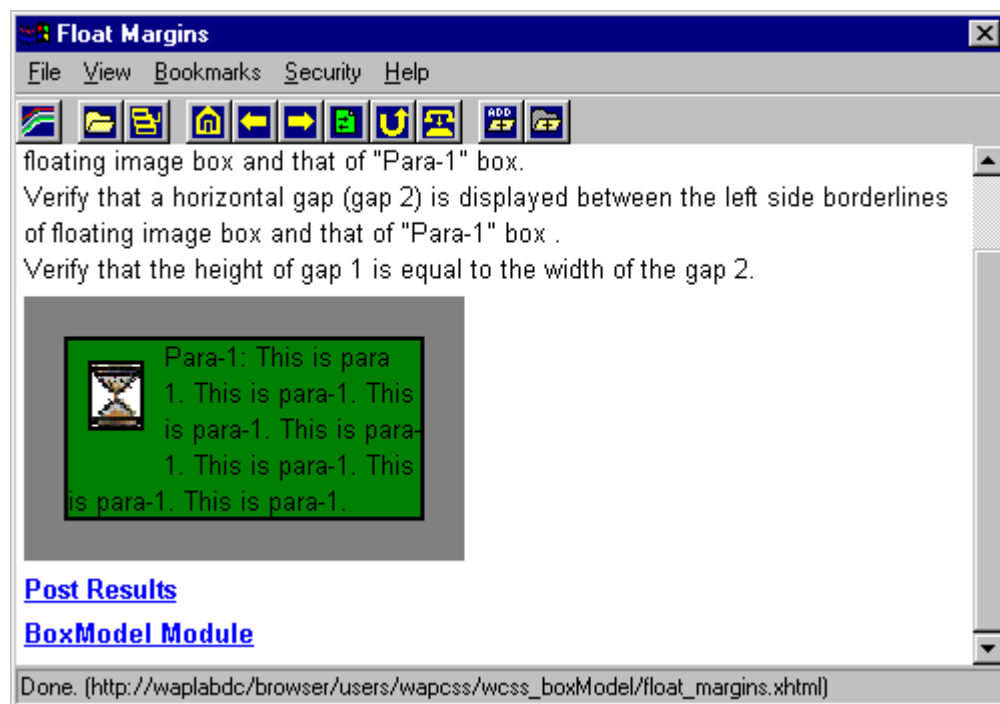
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1 and Section 9.5

Verification: Verify that a vertical gap (gap 1) is displayed between topside borderlines of floating image box and that of "Para-1" box. Verify that a horizontal gap (gap 2) is displayed between the left side borderlines of floating image box and that of "Para-1" box. Verify that the height of gap 1 is equal to the width of the gap 2.



45.19.41 Negative Margins 1

wcss_boxmodel/neg_margins1.xhtml

Assertion : When the MARGIN property is set as negative values for an element in a style sheet, the user agent must apply the rule and the computed values for each instance of the element in the document tree.

Traceability:

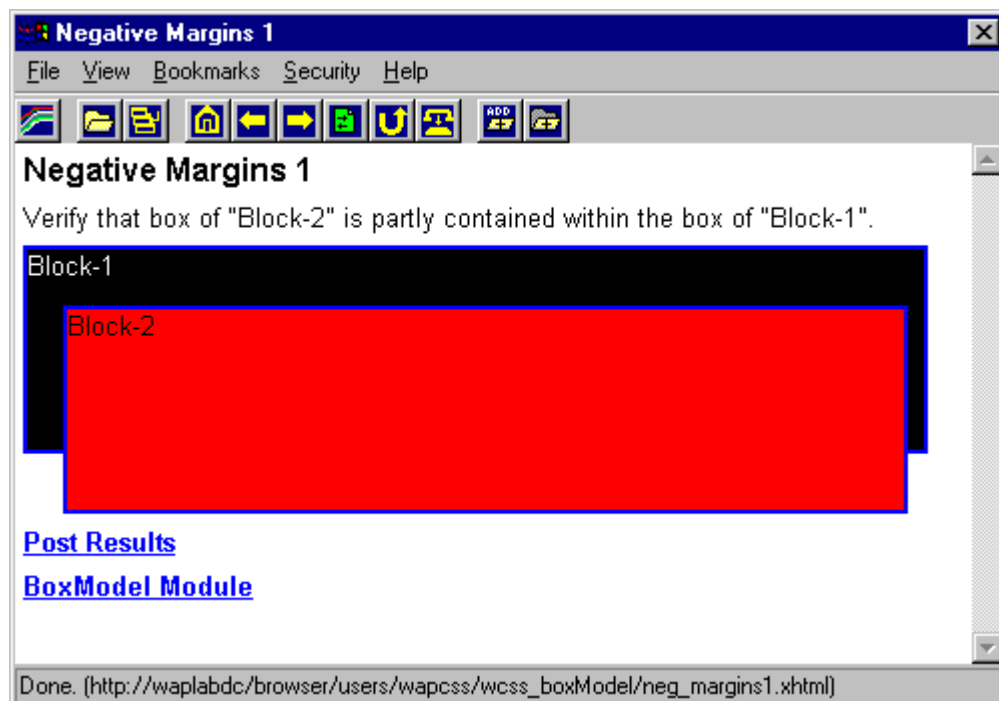
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1

Verification: Verify that box of "Block-2" is partly contained within the box of "Block-1".



45.19.42 Negative Margins 2

wcss_boxmodel/neg_margins2.xhtml

Assertion : When the MARGIN property is set as negative values for an element in a style sheet, the user agent must apply the rule and the computed values for each instance of the element in the document tree.

Traceability:

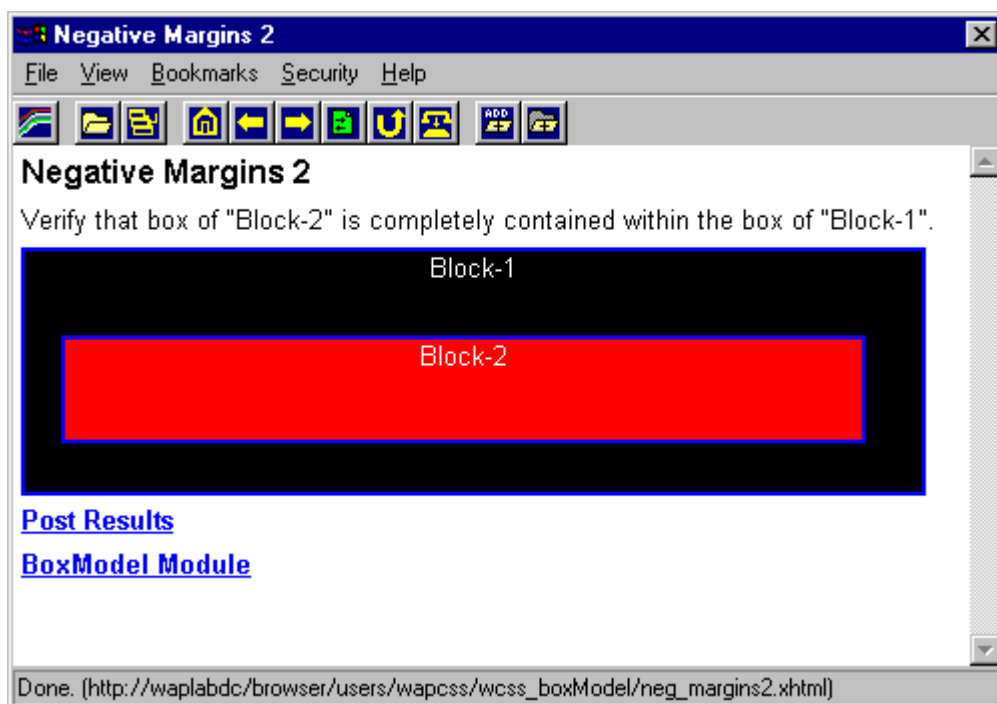
[WAPCSS]

Section 10.1

[CSS2]

Section 8.3.1

Verification: Verify that box of "Block-2" is completely contained within the box of "Block-1".



45.19.43 Invalid Colors

wcss_boxmodel/inv_colors.xhtml

Assertion : When the 'border-color' property is set to an invalid value and when 'border-style' property is set to 'solid' for an element E, then the UA must display solid border lines in default color for that element.

Traceability:

[WAPCSS]

Section 10.4

[CSS2]

Section 8.5.2

Verification: Verify that both "Block-1" and "Block-2" are displayed with solid borderlines of default color .

10 COLORS & BACKGROUND

This section contains tests for the foreground color, background color and background image properties (45.20.01 to 45.20.33).

10.1 Foreground Color

The foreground color properties are tested in this sub-module. The start pages are described below (45.20.01 to 45.20.03). Each test in this section refers to SCR: WCSS-FCOLOR-C-001.

45.20.01 Foreground Color wcss_colors/foreground/fgcolor.xhtml

Assertion 1: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

Assertion 2: When a property that contains a color value in RGB three integer values format is encountered (eg: rgb(255,0,0)), the user agent must support that property value.

Assertion 3: When a property that contains a color value in three hexadecimal characters format is encountered (eg: #f00), the user agent must support that property value.

Assertion 4: When a property that contains a color value in six hexadecimal characters format is encountered (eg: #ff0000), the user agent must support that property value.

Assertion 5: When a property that contains a color value in RGB three percentage values format is encountered (eg: rgb(100%, 0%, 0%)), the user agent must support that property value.

Traceability of above 5 assertions:

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Assertion 6: When the COLOR property is set for an element in a style sheet, the user agent must apply the defined color value to the element's text content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.1

[CSS2]

Section 14.1

Verification: Verify that the text in "Line-1" to "Line-5" is displayed in red colored font.

45.20.02 Color-keywords 1

wcss_colors/foreground/fgcolor_key1.xhtml

Assertion 1: When the COLOR property is set for an element in a style sheet, the user agent must apply the defined color value to the element's text content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.1

[CSS2]

Section 14.1

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

Traceability:

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that the eight color names appear in their actual color in this page (eg: the color name "Blue" should be displayed in blue color).

45.20.03 Color-keywords 2

wcss_colors/foreground/fgcolor_key2.xhtml

Assertion 1: When the COLOR property is set for an element in a style sheet, the user agent must apply the defined color value to the element's text content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.1

[CSS2]

Section 14.1

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

Traceability:

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that the eight color names appear in their actual color in this page (eg: the color name "Red" should be displayed in red color). A black-color background is provided for white and yellow color texts for better visibility of the text.

10.2 Background Color

The background color properties are tested in this sub-module. The start pages are described below (45.20.04). Each test in this section refers to SCR: WCSS-BCOLOR-C-001.

45.20.04 Background Color `wcss_colors/bgcolor/bg_color.xhtml`

Assertion 1: When the BACKGROUND-COLOR property is set for an element in a style sheet, the user agent must apply the defined color value to the background for each instance of the element in the document tree.

Assertion 2: When the BACKGROUND-COLOR property is set to 'transparent' for an element in a style sheet, the user agent must apply the same color value as its parent element to the background for each instance of the element in the document tree.

Assertion 3: When the BACKGROUND-COLOR property is set as a color value (numerical RGB specification) for an element in a style sheet, the user agent must apply the defined color value to the background for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]

Section 11.2

[CSS2]

Section 14.2.1

Verification: Verify that the "Para-1" has red colored background and "Para-2" has green colored background. Verify that "Para-3, Line-1" has red colored background and "Para-3, Line-2" has green colored background.

10.3 Background Images

The properties 'background-image', 'background-repeat', 'background-attachment', 'background-position' and the 'background' shorthand properties are covered in this section. The tests numbered 45.20.05 to 45.20.33 belongs to this section. Each test in this section refers to SCR: WCSS-BIMAGES-C-001.

45.20.05 Background Image wcss_colors/bgimage/bg_image.xhtml

Assertion 1: When the BACKGROUND-IMAGE property is set to 'none' for an element in a style sheet, the user agent must not display any background images for each instance of the element in the document tree.

Assertion 2: When the BACKGROUND-IMAGE property is set as an uri for an element in a style sheet, the user agent must display, if available, the defined image as background for each instance of the element in the document tree.

Assertion 3: When the BACKGROUND-REPEAT property is unset and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if available, the defined image as background and repeat the image both horizontally and vertically for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]	Section 11.3
[CSS2]	Section 14.2.1

Assertion 4: When a property that contains an uri value is encountered, the user agent must support that property value.

Traceability:

[WAPCSS]	Section 6.2
[CSS2]	Section 4.3.4

Verification: Verify that "Para-1" does not have any image in the background. Verify that "Para-2" has a background image, which is repeated in both horizontal and vertical directions in the entire background of the para.

45.20.06 Image Repeat

wcss_colors/bgimage/bg_image_repeat.xhtml

Assertion 1: When the BACKGROUND-REPEAT property is set to 'no-repeat' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if available, only one copy of the defined image as background for each instance of the element in the document tree.

Assertion 2: When the BACKGROUND-POSITION property is unset, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must display the background image, if the image is available, at the top left position of the page for each instance of the element in the document tree.

Assertion 3: When the BACKGROUND-REPEAT property is set to 'repeat' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if available, the defined image as background and repeat the image both horizontally and vertically for each instance of the element in the document tree.

Traceability of above 3 assertions is:

[WAPCSS]
[CSS2]

Section 11.3
Section 14.2.1

Verification: Verify that there is only one single image displayed in the background of "Para-1" placed in the top left corner. Verify that the background image is repeated in both horizontal and vertical directions for entire "Para 2".

45.20.07 Repeat XY

wcss_colors/bgimage/bg_image_repeat_XY.xhtml

Assertion 1: When the BACKGROUND-REPEAT property is set to 'repeat-x' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if available, the defined image as background and repeat the image horizontally only for each instance of the element in the document tree.

Assertion 2: When the BACKGROUND-REPEAT property is set to 'repeat-y' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if available, the defined image as background and repeat the image vertically only for each instance of the element in the document tree.

Traceability of above 2 assertions is:

[WAPCSS]
[CSS2]

Section 11.3
Section 14.2.1

Verification: Verify that the background image is repeated only in the horizontal direction starting from top-left corner in the background of "Para 1". Verify that the background image is repeated only in the vertical direction starting from top-left corner for "Para 2"

45.20.08 **Image Attach** **wcss_colors/bg_attachment/bg_image_attach.xhtml**

Assertion: When the BACKGROUND-ATTACHMENT property is unset and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if the image is available, a background able to scroll at the same speed as the text for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 11.3
Section 14.2.1

Verification: Verify that the entire page has a background-image repeated all over. Verify that when the page is scrolled, the background image also appears to be scrolling along with it.

45.20.09 **Image Scroll** **wcss_colors/bg_attachment/bg_image_attach_scroll.xhtml**

Assertion: When the BACKGROUND-ATTACHMENT property is set to 'scroll' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if the image is available, a background able to scroll at the same speed as the text for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 11.3
Section 14.2.1

Verification: Verify that the entire page has a background-image repeated all over. Verify that when the page is scrolled, the background image also appears to be scrolling along with it.

45.20.10 **Image Fixed** **wcss_colors/bg_attachment/bg_image_attach_fixed.xhtml**

Assertion: When the BACKGROUND-ATTACHMENT property is set to 'fixed' and the BACKGROUND-IMAGE property is set for an element in a style sheet, the user agent must display, if the image is available, a background able to "glue" to the view port when the element is scrolled for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 11.3
Section 14.2.1

Verification: Verify that the entire page has a background-image repeated all over. Verify that when the page is scrolled, the background image appears to be fixed to the view-port.

45.20.11 Image Upper Left

wcss_colors/bg_position/bg_upperleft.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'top left' or 'left top', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the upper left corner of the background image, if the image is available, at the upper left corner of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" and "Para-2" has a single non-repeated image displayed in the background at top-left corner.

45.20.12 Image Upper Center

wcss_colors/bg_position/bg_uppercenter.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'top', 'center top' or 'top center', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the upper left corner of the background image, if the image is available, at the upper center of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" to "Para-3" has a single non-repeated image displayed in the background at upper center portion of the para.

45.20.13 Image Upper Right

wcss_colors/bg_position/bg_upperright.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'top right' or 'right top', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the upper right corner of the background image, if the image is available, at the upper right corner of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" and "Para-2" has a single non-repeated image displayed in the background at top-right corner.

45.20.14 Image Left Center

wcss_colors/bg_position/bg_leftcenter.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'left', 'center left' or 'left center', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the background image, if the image is available, at the left center of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" to "Para-3" has a single non-repeated image displayed in the background at the left center of the para.

45.20.15 Image Center

wcss_colors/bg_position/bg_center.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'center' or 'center center', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the background image, if the image is available, at the center of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" and "Para-2" has a single non-repeated image displayed in the background at the center of the para.

45.20.16 Image Right Center

wcss_colors/bg_position/bg_rightcenter.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'right', 'center right' or 'right center', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the background image, if the image is available, at the right center of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" to "Para-3" has a single non-repeated image displayed at the right center of the para.

45.20.17 Image Bottom Left

wcss_colors/bg_position/bg_bottomleft.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'bottom left' or 'left bottom', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the background image, if the image is available, at the lower left corner of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" and "Para-2" has a single non-repeated image displayed in the background at bottom-left corner.

45.20.18 Image Bottom Center

wcss_colors/bg_position/bg_bottomcenter.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'bottom', 'center bottom' or 'bottom center', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the background image, if the image is available, at the bottom center of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" to "Para-3" has a single non-repeated image displayed in the background at bottom center portion of the para.

45.20.19 Image Bottom Right

wcss_colors/bg_position/bg_bottomright.xhtml

Assertion: When the BACKGROUND-POSITION property is set to 'bottom right' or 'right bottom', the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must place the lower right corner of the background image, if the image is available, at the lower right corner of the element's padding area for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that each of the "Para-1" and "Para-2" has a single non-repeated image displayed in the background at bottom-right corner.

45.20.20 1 Length

wcss_colors/bg_position/bg_1length.xhtml

Assertion: When the BACKGROUND-POSITION property is set as a single length value, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must consider the defined value as the horizontal value and apply the 50% value as the vertical value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the background image in "Para-2" is placed farther from the left edge of the para in horizontal direction, when compared to the placing of background image in "Para-1".

45.20.21 1 Percent

wcss_colors/bg_position/bg_1percent.xhtml

Assertion: When the BACKGROUND-POSITION property is set as a single percentage value, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must consider the defined value as the horizontal value and apply the 50% value as the vertical value and place the image accordingly after computing the position, for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the background image in "Para-2" is placed farther from the left edge of the para in horizontal direction, when compared to the placing of background image in "Para-1".

45.20.22 2 Lengths

wcss_colors/bg_position/bg_2lengths.xhtml

Assertion: When the BACKGROUND-POSITION property is set as two length values, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must consider the first defined value as the horizontal value and the second defined value as the vertical value and place the image accordingly after computing the position, for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the background image in "Para-2" is placed farther from the top-left corner of the para, when compared to the placing of background image in "Para-1".

45.20.23 2 Percents

wcss_colors/bg_position/bg_2percents.xhtml

Assertion When the BACKGROUND-POSITION property is set as two length values, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must consider the first defined value as the horizontal value and the second defined value as the vertical value and place the image accordingly after computing the position, for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the background image in "Para-2" is placed farther from the top-left corner of the para, when compared to the placing of background image in "Para-1".

45.20.24 1 Length – 1 Percent

wcss_colors/bg_position/bg_len_percent.xhtml

Assertion: When the BACKGROUND-POSITION property is set as a percentage and length value pair, the BACKGROUND-IMAGE property is set and the BACKGROUND-REPEAT property is set to 'no-repeat' for an element in a style sheet, the user agent must consider the first defined value as the horizontal value and the second defined value as the vertical value and place the image accordingly after computing the position, for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the background image in "Para-2" is placed farther from the top-left corner of the para, when compared to the placing of background image in "Para-1".

45.20.25 Color

wcss_colors/bg_shand/bgsh_color.xhtml

Assertion: When the BACKGROUND property is set to a color value for an element in a style sheet, the user agent must apply the defined color value to the background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll'), the BACKGROUND-POSITION ('0% 0%'), the BACKGROUND-REPEAT ('repeat') and the BACKGROUND-IMAGE ('none') properties to the background for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the entire page has yellow colored background and there is no background image displayed.

45.20.26 Image

wcss_colors/bg_shand/bgsh_image.xhtml

Assertion: When the BACKGROUND property is set to an URI for an element in a style sheet, the user agent must display the image, if the image is available, as the element's background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll'), the BACKGROUND-POSITION ('0% 0%') and the BACKGROUND-REPEAT ('repeat') properties to the element's background for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that an image is displayed repeated in both x and y directions in the entire background of the page. Verify that the background image scrolls along with the page, when the page is scrolled.

45.20.27 No Repeat

wcss_colors/bg_shand/bgsh_norep.xhtml

Assertion 1: When the BACKGROUND property is set to an URI and 'no-repeat' for an element in a style sheet, the user agent must display the image, if the image is available, as the element's background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll') and the BACKGROUND-POSITION ('0% 0%') properties to the element's background for each instance of the element in the document tree.

Assertion 2: When the BACKGROUND property is set to an URI, a color value and 'no-repeat' for an element in a style sheet, the user agent must display the image, if the image is available, as the element's background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll') and the BACKGROUND-POSITION ('0% 0%') properties to the element's background for each instance of the element in the document tree. The defined color must fill the holes of the background image and fill the rest of the element's padding if the image is too small.

Assertion 3: When the BACKGROUND property is set to an URI, 'center' and 'no-repeat' for an element in a style sheet, the user agent must place one background image, if the image is available, at the center of the element's padding area and apply the initial value of the BACKGROUND-ATTACHMENT ('scroll') property to the element's background for each instance of the element in the document tree.

Traceability of above 3 assertions is:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that "Para-1" has a single non-repeated image displayed at the top-left corner in the background. Verify that "Para-2" also has a single non-repeated image displayed at the top-left corner in the background. Verify that "Para-3" has a single non-repeated image displayed at the center of the para in the background. Verify that both "Para-2" and "Para-3" has yellow colored background displayed.

45.20.28 Fixed 1

wcss_colors/bg_shand/bgsh_fixed1.xhtml

Assertion: When the BACKGROUND property is set to an URI, 'fixed' and 'no-repeat' for an element in a style sheet, the user agent must place one background image, if the image is available, at the upper left corner of the element's padding area (0% 0% as initial value) and display a background able to "glue" to the view port when the element is scrolled for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the page has a background image placed at top left corner. Verify that this background image appears to be fixed to the view-port, when the page is scrolled.

45.20.29 Fixed 2

wcss_colors/bg_shand/bgsh_fixed2.xhtml

Assertion: When the BACKGROUND property is set to an URI, a color value, 'fixed' and 'no-repeat' for an element in a style sheet, the user agent must place one background image, if the image is available, at the upper left corner of the element's padding area (0% 0% as initial value) and display a background able to "glue" to the view port when the element is scrolled for each instance of the element in the document tree. The defined color must fill the holes of the background image and fill the rest of the element's padding if the image is too small.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the page has a background image placed at top left corner. Verify that a blue colored background is displayed at the rest of the page. Verify that the background image appears to be fixed to the view-port, when the page is scrolled.

45.20.30 Fixed 3

wcss_colors/bg_shand/bgsh_fixed3.xhtml

Assertion: When the BACKGROUND property is set to an URI, 'fixed', 'top right' and 'no-repeat' for an element in a style sheet, the user agent must place one background image, if the image is available, at the upper right of the element's padding area and display a background able to "glue" to the view port when the element is scrolled for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Verification: Verify that the page has a background image placed at top right corner. Verify that a blue colored background is displayed at the rest of the page. Verify that the background image appears to be fixed to the view-port, when the page is scrolled.

45.20.31 Named Colors 1

wcss_colors/bg_shand/bgcol_named1.xhtml

Assertion 1: When the BACKGROUND property is set to a color value for an element in a style sheet, the user agent must apply the defined color value to the background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll'), the BACKGROUND-POSITION ('0% 0%'), the BACKGROUND-REPEAT ('repeat') and the BACKGROUND-IMAGE ('none') properties to the background for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that each of the "Block-1" to "Block-5" is displayed with a background color as per the description given within the block.

45.20.32 Named Colors 2

wcss_colors/bg_shand/bgcol_named2.xhtml

Assertion 1: When the BACKGROUND property is set to a color value for an element in a style sheet, the user agent must apply the defined color value to the background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll'), the BACKGROUND-POSITION ('0% 0%'), the BACKGROUND-REPEAT ('repeat') and the BACKGROUND-IMAGE ('none') properties to the background for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that each of the "Block-1" to "Block-5" is displayed with a background color as per the description given within the block.

Assertion 1: When the BACKGROUND property is set to a color value for an element in a style sheet, the user agent must apply the defined color value to the background and apply the initial values of the BACKGROUND-ATTACHMENT ('scroll'), the BACKGROUND-POSITION ('0% 0%'), the BACKGROUND-REPEAT ('repeat') and the BACKGROUND-IMAGE ('none') properties to the background for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 11.3

[CSS2]

Section 14.2.1

Assertion 2: When a property that contains a color (The 16 HTML4.0 colors) value is encountered, the user agent must support that property value.

[WAPCSS]

Section 6.2

[CSS2]

Section 4.3.6

Verification: Verify that each of the "Block-1" to "Block-6" is displayed with a background color as per the description given within the block.

11 FONTS

The properties font-family, font-style, font-size, font-variant, font-weight and the short hand 'font' property are covered in this module.

11.1 Font Style

This sub-module contains test cases of "font-style" property. The start pages of test cases are described below (45.21.01 and 45.21.02). Each test in this section refers to SCR: WCSS-FONTS-STYLE-C-001.

45.21.01 Italic Font

wcss_fonts/fontstyle/fs_italic.xhtml

Assertion 1: When the FONT-STYLE property is set to 'italic' for an element in a style sheet, the user agent must use the font classified as italic in the UA's font database for each instance of the element in the document tree.

Assertion 2: When the FONT-STYLE property is unset for an element in a style sheet, the user agent must use the font classified as 'normal' in the UA's font database for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]

Section 12.2

[CSS2]

Section 15.2.3

Verification: Verify that the "Para 2" is displayed in italic, not bold font and "Para 1" is displayed in normal font.

45.21.02 Oblique Font

wcss_fonts/fontstyle/fs_oblique.xhtml

Assertion 1: When the FONT-STYLE property is set to 'oblique' for an element in a style sheet, the user agent must use the font classified as oblique in the UA's font database for each instance of the element in the document tree.

Assertion 2: When the FONT-STYLE property is set to 'normal' for an element in a style sheet, the user agent must use the font classified as normal in the UA's font database for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]

Section 12.2

[CSS2]

Section 15.2.3

Verification: Verify that the "Para 2" is displayed in oblique, not bold font and "Para 1" is displayed in normal font.

11.2 Font Variant

This sub-module contains test cases of "font-variant" property. The start pages are described below (45.21.03). Each test in this section refers to SCR: WCSS-FONTS-VARIANT-C-001.

45.21.03 Small Caps

wcss_fonts/fontvariant/fv_scaps.xhtml

Assertion 1: When the FONT-VARIANT property is set to 'small-caps' for an element in a style sheet, the user agent must use a font that is labeled as a small-caps font or simulate the small-caps font by using scaled uppercase characters or as a last resort, use unscaled uppercase characters for each instance of the element in the document tree.

Assertion 2: When the FONT-VARIANT property is set to 'normal' for an element in a style sheet, the user agent must use a font that is NOT labeled as a small-caps font for each instance of the element in the document tree.

Assertion 3: When the FONT-VARIANT property is unset for an element in a style sheet, the user agent must use a font that is NOT labeled as a small-caps font for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]
[CSS2]

Section 12.3
Section 15.2.3

Verification: Verify that the entire text in "Para 1" is displayed in small-caps font or in regular upper case characters. Small-caps font is upper case characters in slightly smaller size than regular upper case characters. Verify that text in "Para 2" and "Para 3" is neither displayed in small-caps font nor in regular upper case characters.

Eg: THIS IS SMALL CAPS TEXT.
 THIS IS REGULAR UPPER CASE TEXT.

11.3 Font Weight

This sub-module contains test cases of "font-weight" property. The start pages are described below (45.21.04 to 45.21.10). Each test in this section refers to SCR: WCSS-FONTS-WEIGHT-C-001.

45.21.04 Numeric Weight Font wcss_fonts/fontweight/fw_numeric.xhtml

Assertion: When the FONT-WEIGHT property is set to a numeric value ('100', '200', '300', '400', '500', '600', '700', '800', '900') for an element in a style sheet, the user agent must display the weight of the element's font with the defined value for each instance of the element in the document tree; the numeric values form an ordered sequence, where each number indicates a weight that is at least as dark as its predecessor.

Traceability:

[WAPCSS]

Section 12.4

[CSS2]

Section 15.2.3

Verification: Verify that each of the "Line 1" to "Line 9" are displayed in equally dark font or in darker font than its predecessor, but not in a lighter font than its predecessor.

45.21.05 Normal Weight Font wcss_fonts/fontweight/fw_normal.xhtml

Assertion 1: When the FONT-WEIGHT property is set to 'normal' for an element in a style sheet, the user agent must display the weight of the element's font equal to the numeric value '400' for each instance of the element in the document tree.

Assertion 2: When the FONT-WEIGHT property is unset for an element in a style sheet, the user agent must display the weight of the element's font equal to the numeric value '400' for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]

Section 12.4

[CSS2]

Section 15.2.3

Verification: Verify that each of the "Line 1" to "Line 3" is displayed in normal weight font (neither darker nor lighter than the regular text).

45.21.06 Bold Font

wcss_fonts/fontweight/fw_bold.xhtml

Assertion: When the FONT-WEIGHT property is set to 'bold' for an element in a style sheet, the user agent must display the weight of the element's font equal to the numeric value '700' for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.4

[CSS2]

Section 15.2.3

Verification: Verify that both "Line 1" and "Line 2" are displayed in bold font (darker than normal font).

45.21.07 Bolder Font 1

wcss_fonts/fontweight/fw_bolder1.xhtml

Assertion: When the FONT-WEIGHT property is set to 'bolder' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is NOT equal to '900', the user agent must display the weight of the element's font darker than the inherited one for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.4

[CSS2]

Section 15.2.3

Verification: Verify that in each of the "Parent-Child" pairs 1 to 5, the text of the "Child" line appears in equally dark or darker font than that of its "Parent" line. (Eg: "Child 3" appears in a darker font or equally dark font as that of "Parent 3", but not in a lighter font than "Parent 3")

45.21.08 Bolder Font 2

wcss_fonts/fontweight/fw_bolder2.xhtml

Assertion 1: When the FONT-WEIGHT property is set to 'bolder' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is NOT equal to '900', the user agent must display the weight of the element's font darker than the inherited one for each instance of the element in the document tree.

Assertion 2: When the FONT-WEIGHT property is set to 'bolder' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is equal to '900', the user agent must display the weight of the element's equal to '900' for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]

Section 12.4

[CSS2]

Section 15.2.3

Verification: Verify that in each of the "Parent-Child" pairs 6 to 10, the text of the "Child" line appears in equally dark or darker font than that of its "Parent" line. (Eg: "Child 7" appears in a darker font or equally dark font as that of "Parent 7", but not in a lighter font than "Parent 7"). Verify that the text of "Child 11" appears in equally dark font as that of "Parent 11", not more, not less.

45.21.09 Lighter Font 1

wcss_fonts/fontweight/fw_lighter1.xhtml

Assertion 1: When the FONT-WEIGHT property is set to 'lighter' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is NOT equal to '100', the user agent must display the weight of the element's font lighter than the inherited one for each instance of the element in the document tree.

Assertion 2: When the FONT-WEIGHT property is set to 'lighter' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is equal to '100', the user agent must display the weight of the element's equal to '100' for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]
[CSS2]

Section 12.4
Section 15.2.3

Verification: Verify that in each of the "Parent-Child" pairs 2 to 5, the text of the "Child" line appears in equally dark or in a lighter font than that of its "Parent" line. (Eg: "Child 3" appears in a lighter font than or in equally dark font as that of "Parent 3", but not in a darker font than "Parent 3"). Verify that the text of "Child 1" appears in equally dark font as that of "Parent 1", not more, not less.

45.21.10 Lighter Font 2

wcss_fonts/fontweight/fw_lighter2.xhtml

Assertion: When the FONT-WEIGHT property is set to 'lighter' for an element in a style sheet and the FONT-WEIGHT property of the parent's element is NOT equal to '100', the user agent must display the weight of the element's font lighter than the inherited one for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 12.4
Section 15.2.3

Verification: Verify that in each of the "Parent-Child" pairs 6 to 11, the text of the "Child" line appears in equally dark font or in a lighter font than that of its "Parent" line. (Eg: "Child 3" appears in a lighter font than or in equally dark font as that of "Parent 3", but not in a darker font than "Parent 3").

11.4 Font Size

This sub-module contains test cases for "font-size" property. The start pages are described below (45.21.11 to 45.21.15). Each test in this section refers to SCR: WCSS-FONTS-SIZE-C-001.

45.21.11 Absolute Size

wcss_fonts/fontsize/fs_abs.xhtml

Assertion: When the FONT-SIZE property is set to an absolute-size value ('xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large') for an element in a style sheet, the user agent must display the size of the element's font with the defined value for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.5

[CSS2]

Section 15.2.4

Verification: Verify that each of the "Line 1" to "Line 7" is displayed in such a way that its text appears in bigger font than or in equal size font as that of its predecessor. (Eg: "Line 4" appears in bigger font than or in equal size font as "Line 3", but not in a smaller size font than "Line 3")

45.21.12 Medium Size

wcss_fonts/fontsize/fs_medium.xhtml

Assertion: When the FONT-SIZE property is unset for an element in a style sheet, the user agent must display the size of the element's font equal to 'medium' for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.5

[CSS2]

Section 15.2.4

Verification: Verify that the words "Word-1" and "Word-2" are displayed in same size font.

45.21.13 Relative Size

wcss_fonts/fontsize/fs_relative.xhtml

Assertion 1: When the FONT-SIZE property is set to 'larger' (relative-size value) for an element in a style sheet, the user agent must display the size of the element's font larger than the inherited one for each instance of the element in the document tree.

Assertion 2: When the FONT-SIZE property is set to 'smaller' (relative-size value) for an element in a style sheet, the user agent must display the size of the element's font smaller than the inherited one for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]
[CSS2]

Section 12.5
Section 15.2.4

Verification: Verify that the "Word-2" appears in bigger font size than "Word-1". Verify that the "Word-4" appears in smaller font size than "Word-3".

45.21.14 Size in Length

wcss_fonts/fontsize/fs_length.xhtml

Assertion: When the FONT-SIZE property is set as a length value for an element in a style sheet, the user agent must display the size of the element's font with the defined value (as an absolute font size independent of the user agent's font table) for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 12.5
Section 15.2.4

Verification: Verify that the "Word-2" appears in bigger font size than "Word-1". Verify that the "Word-4" appears in smaller font size than "Word-3".

45.21.15 Size in Percent

wcss_fonts/fontsize/fs_percent.xhtml

Assertion: When the FONT-SIZE property is set as a percentage value for an element in a style sheet, the user agent must display the size of the element's font with the defined value (as an absolute font size relative to its parent's element's font size) for each instance of the element in the document tree.

Traceability:

[WAPCSS]
[CSS2]

Section 12.5
Section 15.2.4

Verification: Verify that the "Word-2" appears in bigger font size than "Word-1". Verify that the "Word-4" appears in smaller font size than "Word-3".

11.5 Shorthand Font Property

This sub-module contains test cases for "font" shorthand property. Start pages are described below (45.21.16 to 45.21.19). Each test in this section refers to SCR: WCSS-FONTS-SHORT-C-001.

45.21.16 Set Variant wcss_fonts/font_shand/fvariant.xhtml

Assertion: When the FONT property is set to 'small-caps' for an element in a style sheet, the user agent must use a font that is labeled as a small-caps font or simulate the small-caps font, if not available, and all the initial values of the other fonts properties (FONT-SIZE 'medium', FONT-WEIGHT 'normal', and FONT-STYLE 'normal') for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 12.6
[CSS2]	Section 15.2.5

Verification: Verify that both "Para 1" and "Para 2" are displayed in small-caps font, normal font-size, normal font-style and normal font-weight.

45.21.17 Set Weight wcss_fonts/font_shand/fweight.xhtml

Assertion: When the FONT property is set to a font-weight value for an element in a style sheet, the user agent must display the weight of the element's font equal to the defined value and use all the initial values of the other fonts properties (FONT-SIZE 'medium', FONT-VARIANT 'normal' and FONT-STYLE 'normal') for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 12.6
[CSS2]	Section 15.2.5

Verification: Verify that both "Para 1" and "Para 2" are displayed in normal font-size, normal font-style and darker font.

45.21.18 Set Size

wcss_fonts/font_shand/fsize.xhtml

Assertion: When the FONT property is set to a font-size value for an element in a style sheet, the user agent must display the size of the element's font with the defined value and use all the initial values of the other fonts properties (FONT-WEIGHT 'normal', FONT-VARIANT 'normal' and FONT-STYLE 'normal') for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.6

[CSS2]

Section 15.2.5

Verification: Verify that both "Para 1" and "Para 2" are displayed in bigger font-size, normal font-style and normal font-weight.

45.21.19 Set Style

wcss_fonts/font_shand/fstyle.xhtml

Assertion: When the FONT property is set to a font-style value for an element in a style sheet, the user agent must use the font classified as the defined value in the UA's font database and all the initial values of the other fonts properties (FONT-SIZE 'medium', FONT-WEIGHT 'normal', FONT-VARIANT 'normal') for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 12.6

[CSS2]

Section 15.2.5

Verification: Verify that both "Para 1" and "Para 2" are displayed normal font-size, italic font-style and normal font-weight.

11.6 Font Family

This sub-module contains test cases for "font-family" property. Start pages are described below (45.21.20 to 45.21.21). Each test in this section refers to SCR: WCSS-FONTS-FAMILY-C-001.

45.21.20 Font family 1 wcss_fonts/font_family/ffamily1.xhtml

Assertion 1: When the FONT-FAMILY property is set as a generic family value for an element in a style sheet, the user agent must use the defined font for the element's textual content for each instance of the element in the document tree.

Assertion 2: When the FONT-FAMILY property is set as a family name for an element in a style sheet, the user agent must use the defined font for the element's textual content for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 12.1
[CSS2]	Section 15.2.2

Verification: Verify that page loads without any errors. The text of "Para-1" and "Para-2" may be displayed in different types of fonts.

45.21.21 Font family 2 wcss_fonts/font_family/ffamily2.xhtml

Assertion: When the FONT-FAMILY property is set as a generic family value for an element in a style sheet, the user agent must use the defined font for the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 12.1
[CSS2]	Section 15.2.2

Verification: Verify that page loads without any errors. The text of "Para-1" and "Para-2" may be displayed in different types of fonts.

12.1 LISTS

45.22.01 **Discs** [wcss_lists/lis_type/discs.xhtml](#)

Verification: Verify that every list-item of "List 1" to "List 3" has a disc item displayed as marker.

Assertion: When the LIST-STYLE-TYPE property is set to 'circle' for an list-item element in a style sheet, the user agent must display a circle item in front of the element's textual content for each instance of the element in the document tree.

[WAPCSS] Section 13.2
[CSS2] Section 12.6.2

Verification: Verify that every list-item of "List 1" and "List 2" has a circle item displayed as marker.

45.22.03 Squares

wcss_lists/ls_type/squares.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'square' for an list-item element in a style sheet, the user agent must display a square item in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that every list-item of "List 1" and "List 2" has a square item displayed as marker.

45.22.04 No Markers

wcss_lists/ls_type/type_none.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'none' for an list-item element in a style sheet, the user agent must display no items in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that none of the list-items of "List 1" and "List 2" has any markers displayed.

45.22.05 Decimals

wcss_lists/ls_type/decimals.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'decimal' for an list-item element in a style sheet, the user agent must display an incremental decimal number in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that for each of the "List 1" and "List 2", incremental decimal numbers starting with "1" are displayed as markers for list-items. (Eg: 1,2,3,4)

45.22.06 Lower-Roman

wcss_lists/ls_type/lower_roman.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'lower-roman' for an list-item element in a style sheet, the user agent must display an incremental lower-roman decimal number in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that for each of the "List 1" and "List 2", incremental lowercase roman numbers are displayed as markers for list-items. (Eg: i, ii, iii, iv)

45.22.07 Upper-Roman

wcss_lists/ls_type/upper_roman.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'upper-roman' for an list-item element in a style sheet, the user agent must display an incremental upper-roman decimal number in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that for each of the "List 1" and "List 2", incremental uppercase roman numbers are displayed as markers for list-items. (Eg: I, II, III, IV)

45.22.08 Lower-Alpha

wcss_lists/ls_type/lower_alpha.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'lower-alpha' for an list-item element in a style sheet, the user agent must display a lowercase incremental alphabetic character in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that for each of the "List 1" and "List 2", incremental lowercase ASCII letters are displayed as markers for list-items. (Eg: a, b, c, d).

45.22.09 Upper-Alpha

wcss_lists/ls_type/upper_alpha.xhtml

Assertion: When the LIST-STYLE-TYPE property is set to 'upper-alpha' for an list-item element in a style sheet, the user agent must display an uppercase incremental alphabetic character in front of the element's textual content for each instance of the element in the document tree.

Traceability:

[WAPCSS]

Section 13.2

[CSS2]

Section 12.6.2

Verification: Verify that for each of the "List 1" and "List 2", incremental ASCII letters are displayed as markers for list-items. (Eg: A, B, C, D...).