WCSS Tests

TABLE OF CONTENTS

1 INTRODUCTION

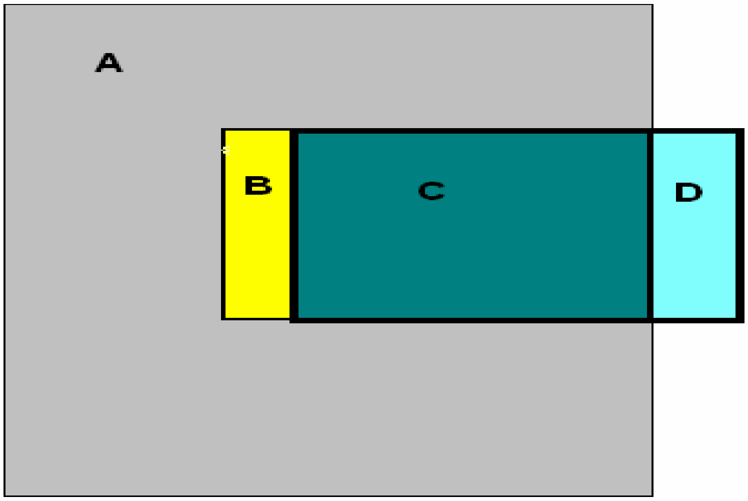
This test plan establishes the guidelines for testing the user-agent's support to the *WAP Cascading Style Sheets* specification [WAPCSS] using WAPCSS test suite. The intention of this testing is to find out the extent of user-agent's support of this specification, but not to verify the strict conformance 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) where as [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 is 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.

Test #	Test Name	Starting Page
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]	Section 6.1
[CSS2]	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]	Section 6.1
[CSS2]	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]	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 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** **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**

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**

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 recurringly 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 DOCUMNET

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**

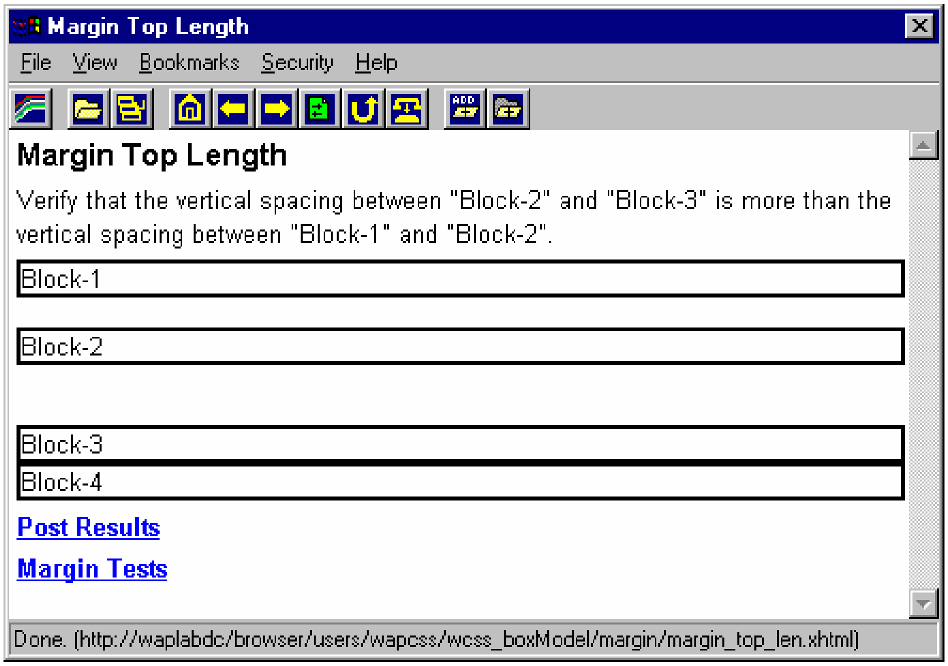
Assertion: When the MARGIN-TOP property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the top margin) and the computed value for each instance of the element in the document tree.

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:



Margin Top Length

Verify that the vertical spacing between "Block-2" and "Block-3" is more than the vertical spacing between "Block-1" and "Block-2".

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_top_len.xhtml)

45.19.02 Margin Top Percent **wcss_boxmodel/margin/margin_top_per.xhtml**

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:

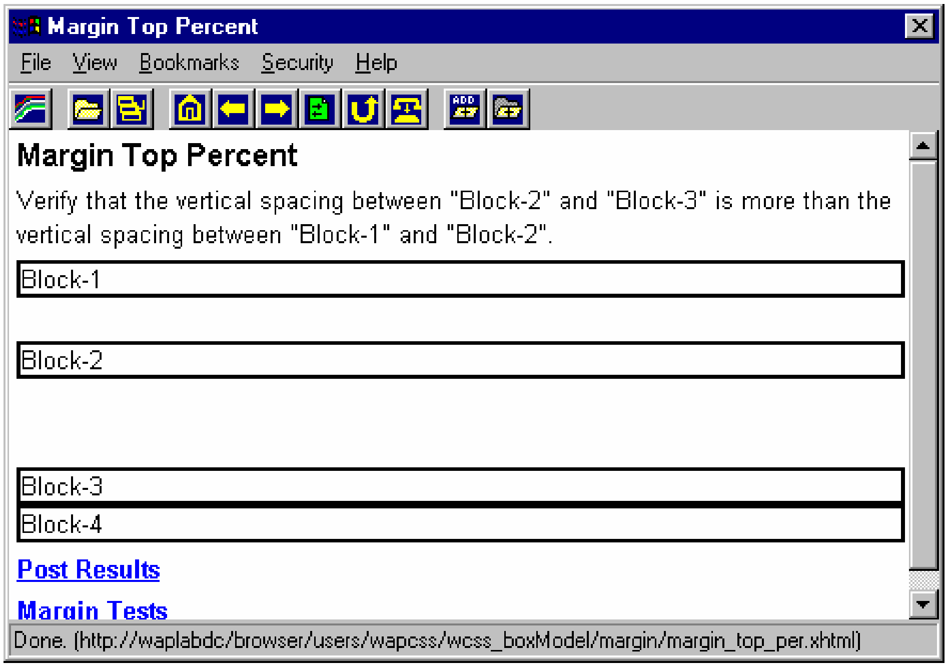
[WAPCSS]	Section 10.1
[CSS2]	Section 8.3

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:



Margin Top Percent

Verify that the vertical spacing between "Block-2" and "Block-3" is more than the vertical spacing between "Block-1" and "Block-2".

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_top_per.xhtml)

45.19.03 Margin Bottom Length **wcss_boxmodel/margin/margin_bot_len.xhtml**

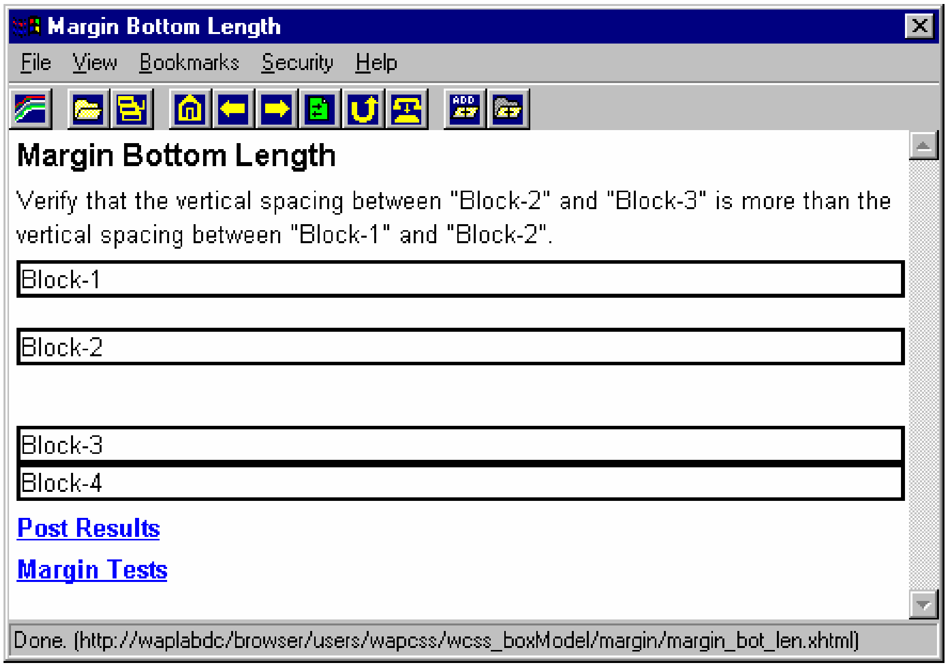
Assertion: When the MARGIN-BOTTOM property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the bottom margin) and the computed value for each instance of the element in the document tree.

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:



Margin Bottom Length

Verify that the vertical spacing between "Block-2" and "Block-3" is more than the vertical spacing between "Block-1" and "Block-2".

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_bot_len.xhtml)

45.19.04 **Margin Bottom Percent** **wcss_boxmodel/margin/margin_bot_per.xhtml**

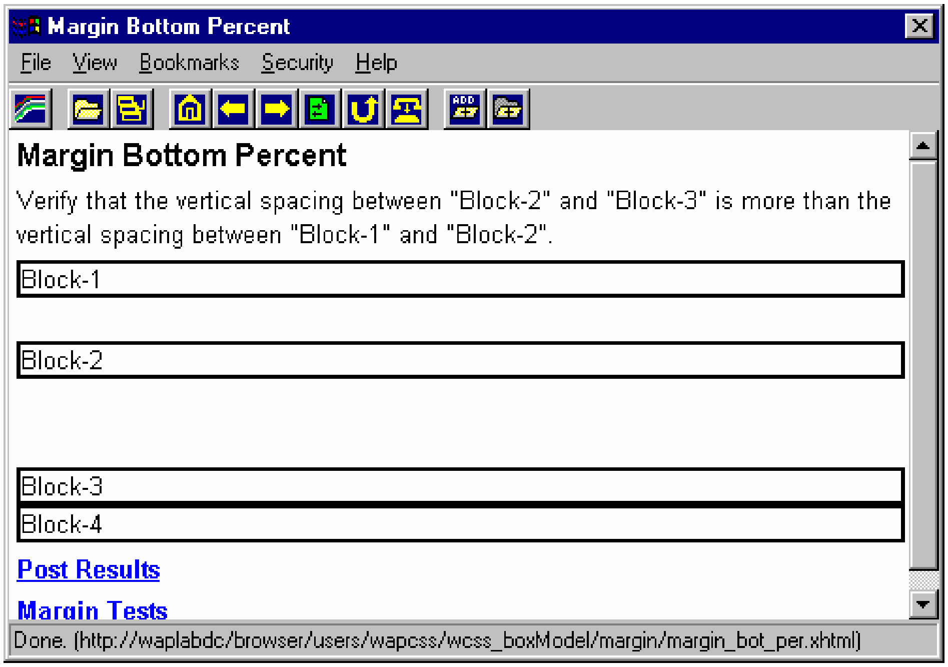
Assertion: When the, MARGIN-BOTTOM property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the bottom margin) and the computed value for each instance of the element in the document tree.

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:



Margin Bottom Percent

Verify that the vertical spacing between "Block-2" and "Block-3" is more than the vertical spacing between "Block-1" and "Block-2".

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_bot_per.xhtml)

45.19.05 Margin Left Length wcss_boxmodel/margin/margin_left_len.xhtml

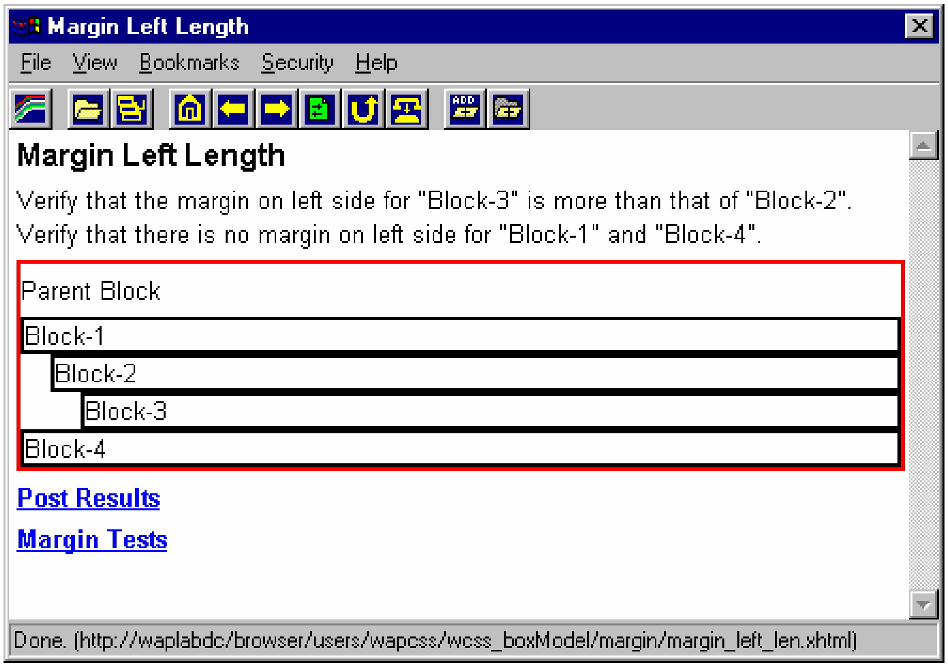
Assertion: When the MARGIN-LEFT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the left margin) and the computed value for each instance of the element in the document tree.

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:



Margin Left Length

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

Parent Block

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_left_len.xhtml)

45.19.06 Margin Left Percent wcss_boxmodel/margin/margin_left_per.xhtml

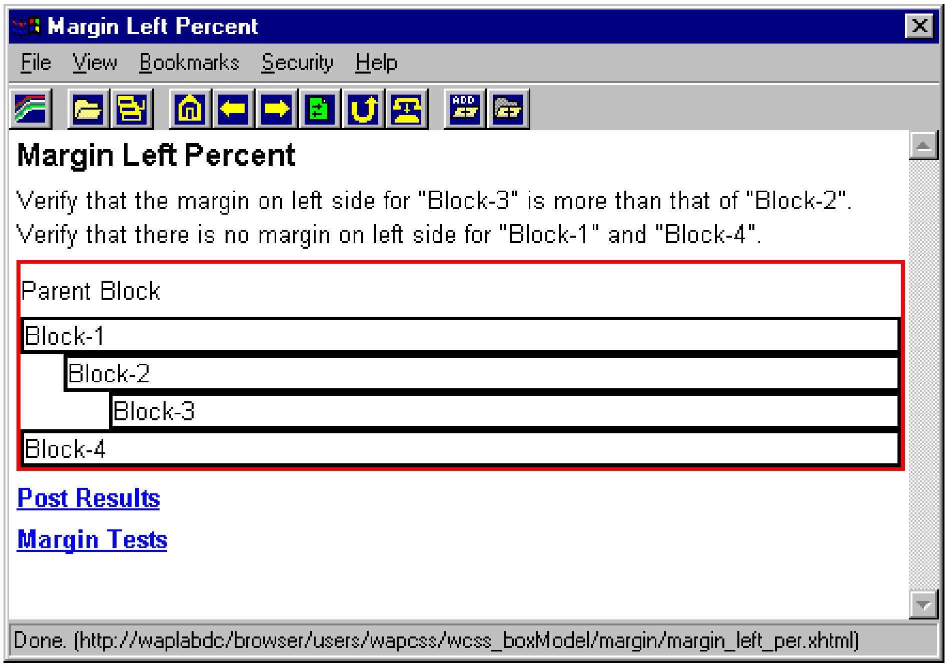
Assertion: When the, MARGIN-LEFT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the left margin) and the computed value for each instance of the element in the document tree.

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:



Margin Left Percent

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

Parent Block

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_left_per.xhtml)

45.19.07 Margin Right Length wcss_boxmodel/margin/margin_right_len.xhtml

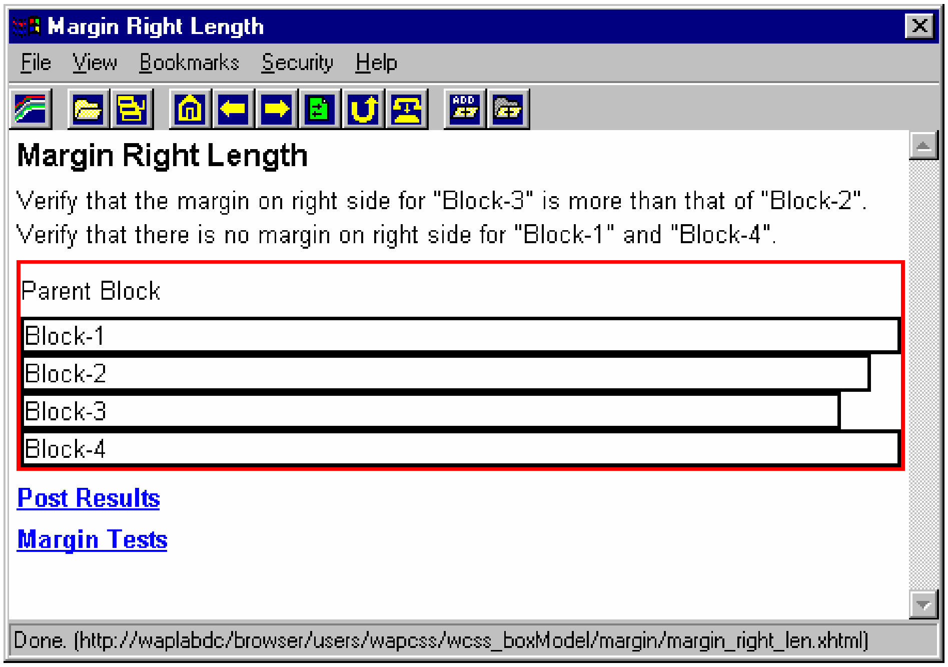
Assertion: When the MARGIN-RIGHT property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the right margin) and the computed value for each instance of the element in the document tree.

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:



Margin Right Length

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".

Parent Block

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_right_len.xhtml)

45.19.08 Margin Right Percent wcss_boxmodel/margin/margin_right_per.xhtml

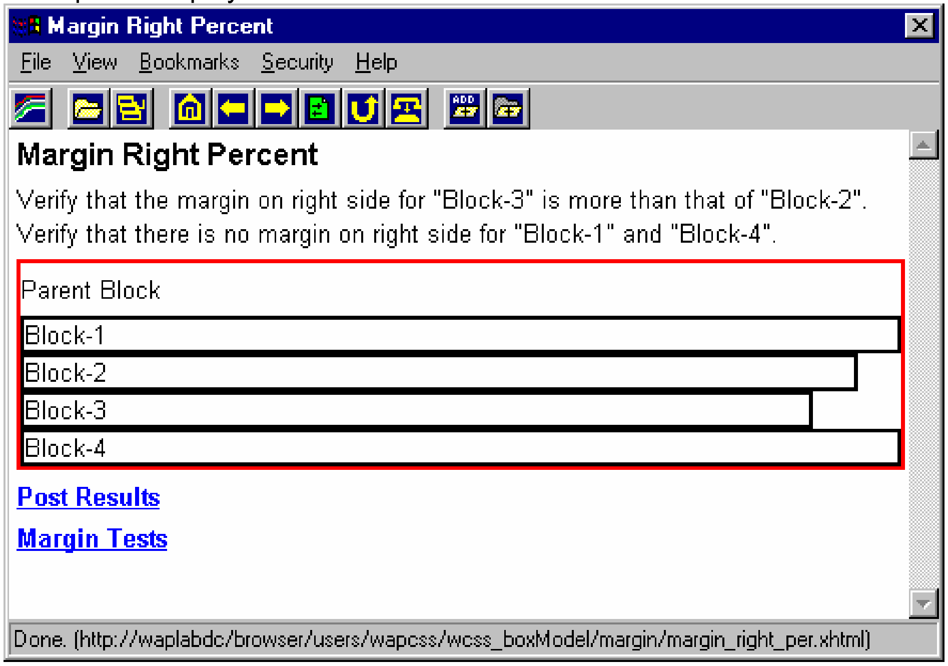
Assertion: When the, MARGIN-RIGHT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the right margin) and the computed value for each instance of the element in the document tree.

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:



Margin Right Percent

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".

Parent Block

Block-1

Block-2

Block-3

Block-4

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_right_per.xhtml)

45.19.09 Margin Shorthand wcss_boxmodel/margin/margin_sh.xhtml

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.

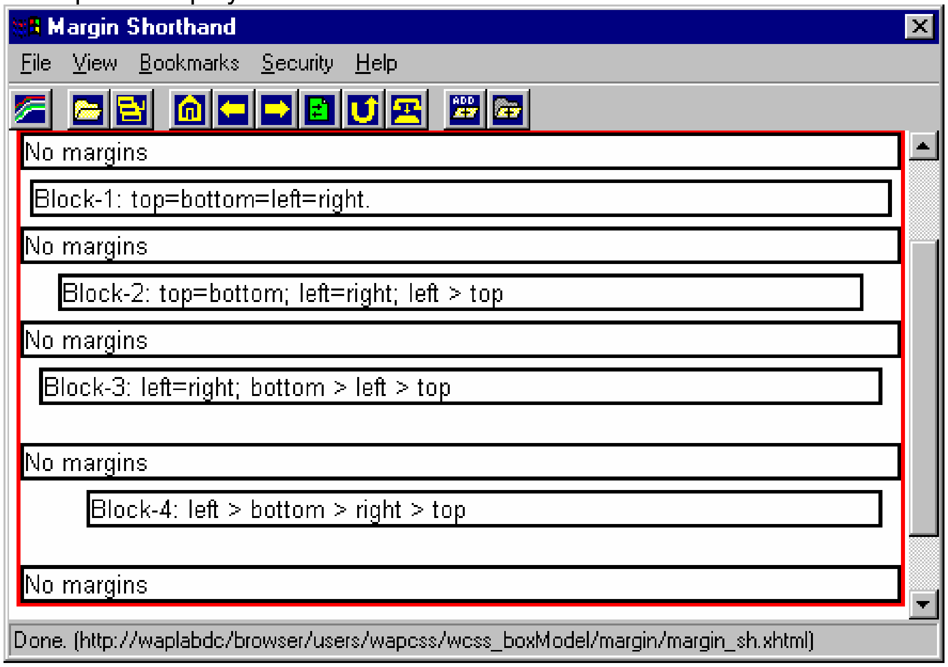
Traceability for all of above 4 assertions is:

[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:



No margins

Block-1: top=bottom=left=right.

No margins

Block-2: top=bottom; left=right; left > top

No margins

Block-3: left=right; bottom > left > top

No margins

Block-4: left > bottom > right > top

No margins

Done. [http://waplabdc/browser/users/wapcss/wcss_boxModel/margin/margin_sh.xhtml]

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

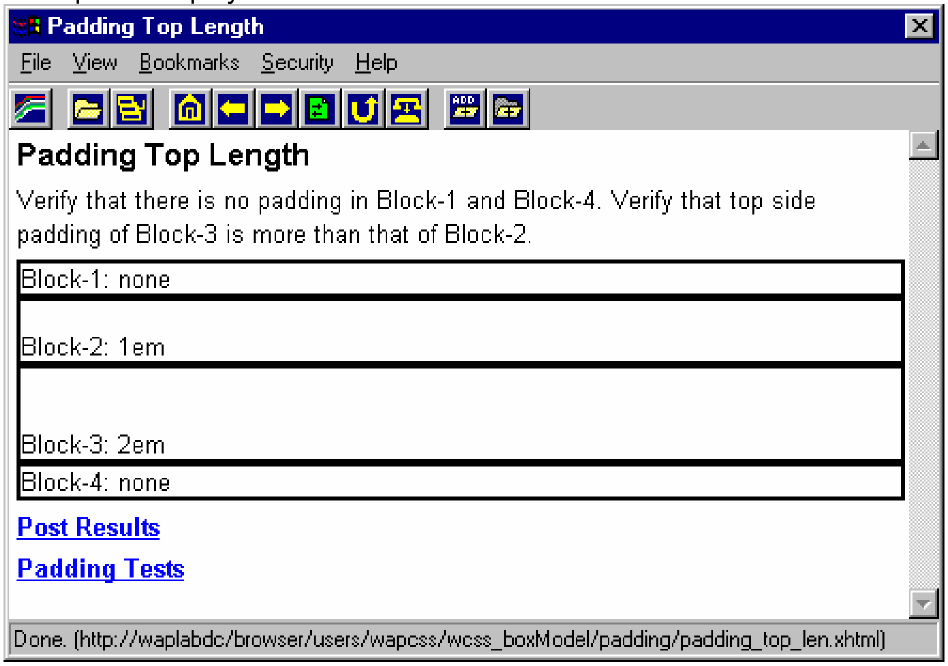
Assertion: When the PADDING-TOP property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the top padding) and the defined value for each instance of the element in the document tree.

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:



Padding Top Length

Verify that there is no padding in Block-1 and Block-4. Verify that top side padding of Block-3 is more than that of Block-2.

Block-1: none

Block-2: 1em

Block-3: 2em

Block-4: none

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_top_len.xhtml)

45.19.11 Padding Top Percent wcss_boxmodel/padding/padding_top_per.xhtml

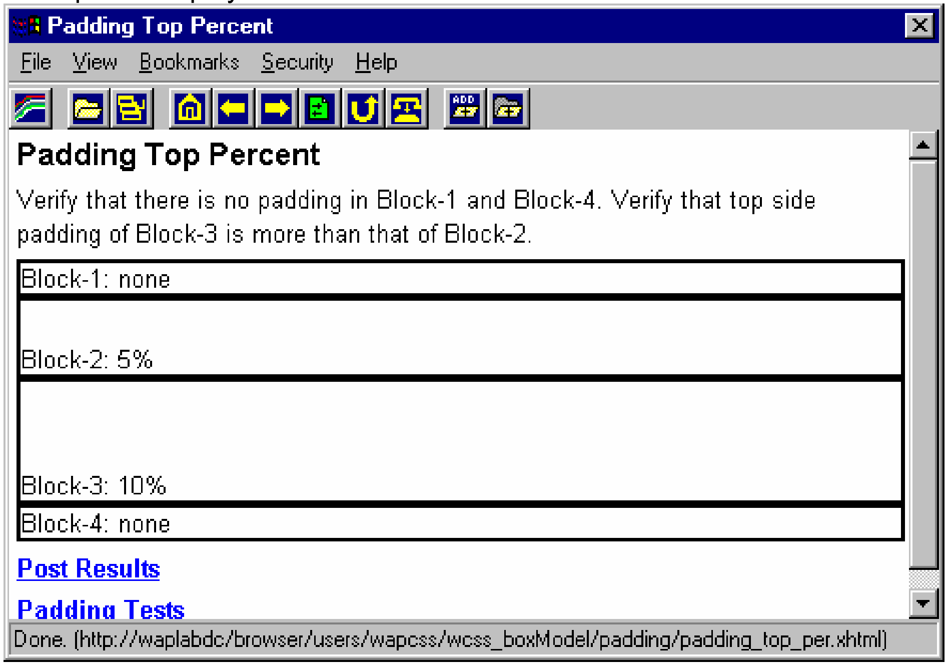
Assertion: When the PADDING-TOP property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the top padding) and the computed value for each instance of the element in the document tree.

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:



Padding Top Percent

Verify that there is no padding in Block-1 and Block-4. Verify that top side padding of Block-3 is more than that of Block-2.

Block-1: none

Block-2: 5%

Block-3: 10%

Block-4: none

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_top_per.xhtml)

45.19.12 Padding Bottom Length **wcss_boxmodel/padding/padding_bot_len.xhtml**

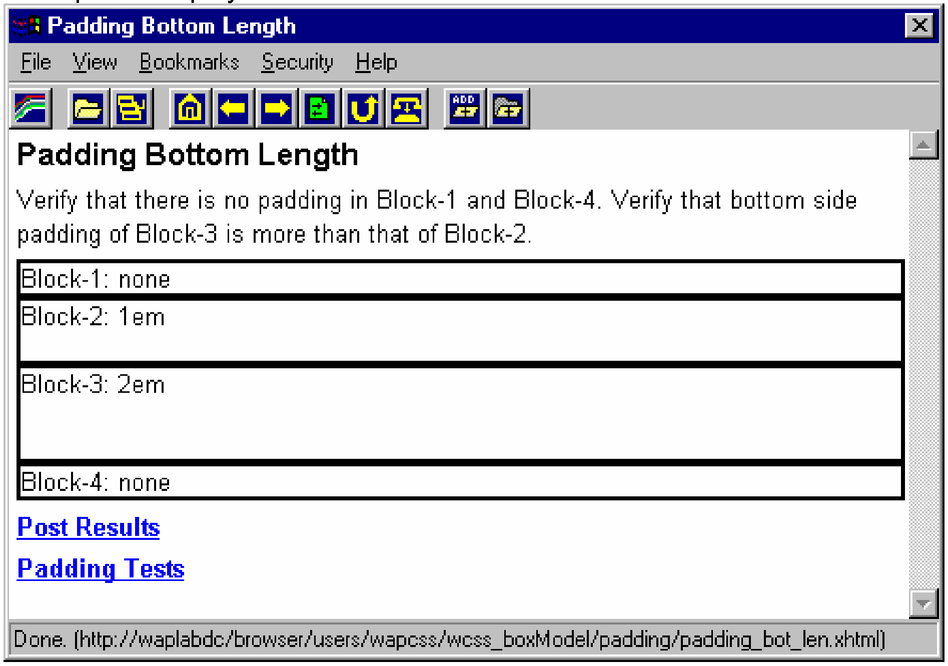
Assertion: When the PADDING-BOTTOM property is set as a <length> value for an element in a style sheet, the user agent must apply the rule (shift the bottom padding) and the defined value for each instance of the element in the document tree.

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:



Padding Bottom Length

Verify that there is no padding in Block-1 and Block-4. Verify that bottom side padding of Block-3 is more than that of Block-2.

Block-1: none

Block-2: 1em

Block-3: 2em

Block-4: none

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_bot_len.xhtml)

45.19.13 **Padding Bottom Percent** **wcss_boxmodel/padding/padding_bot_per.xhtml**

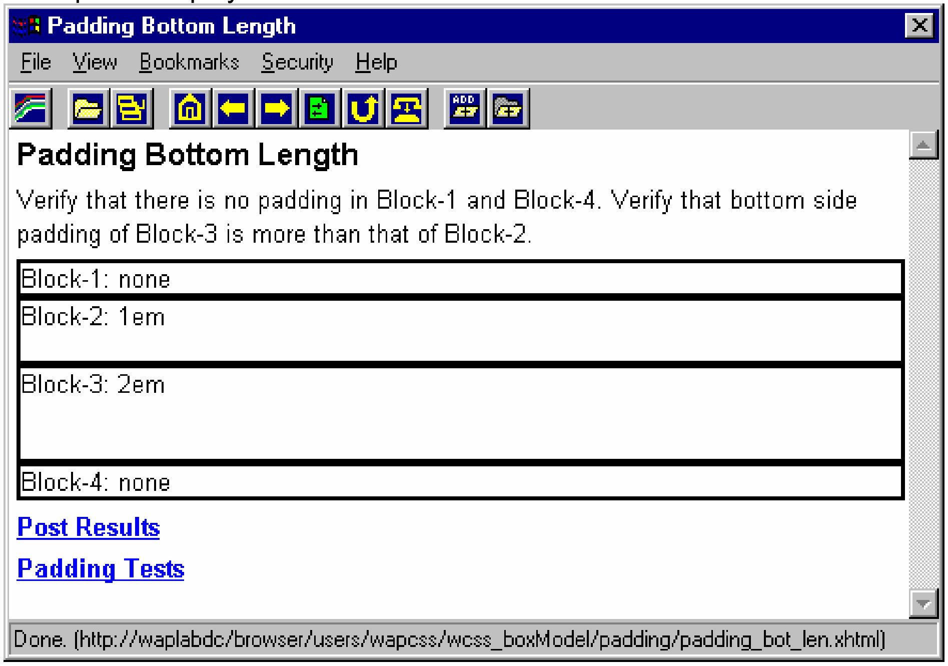
Assertion: When the PADDING-BOTTOM property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the bottom padding) and the computed value for each instance of the element in the document tree.

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:



Padding Bottom Length

Verify that there is no padding in Block-1 and Block-4. Verify that bottom side padding of Block-3 is more than that of Block-2.

Block-1: none

Block-2: 1em

Block-3: 2em

Block-4: none

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_bot_len.xhtml)

45.19.14 Padding Left **wcss_boxmodel/padding/padding_left.xhtml**

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.

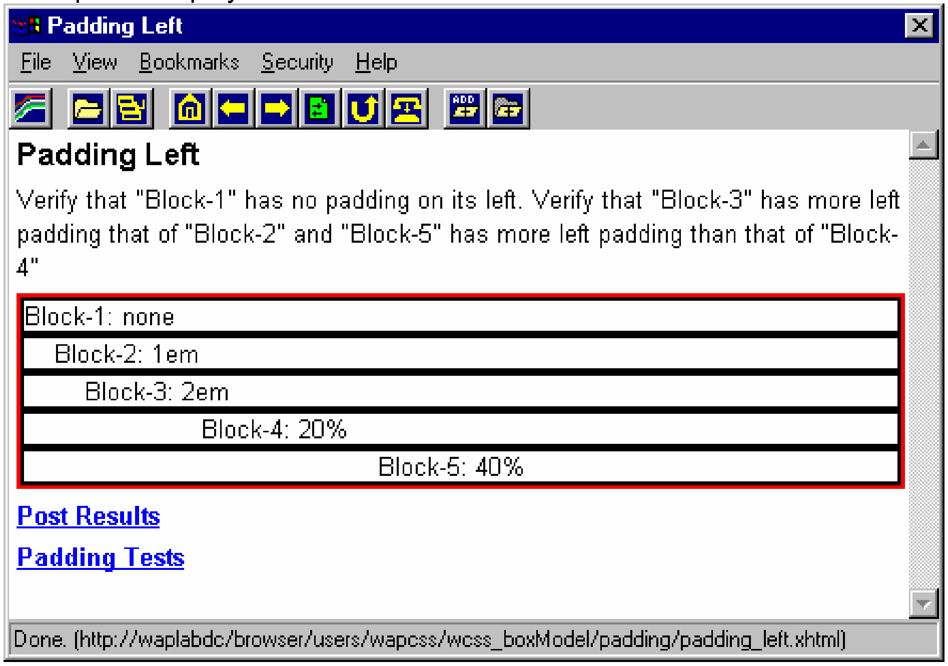
Assertion 2: When the PADDING-LEFT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the left padding) and the computed value for each instance of the element in the document tree.

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:



Padding Left

Verify that "Block-1" has no padding on its left. Verify that "Block-3" has more left padding that of "Block-2" and "Block-5" has more left padding than that of "Block-4"

Block-1: none

Block-2: 1em

Block-3: 2em

Block-4: 20%

Block-5: 40%

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_left.xhtml)

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.

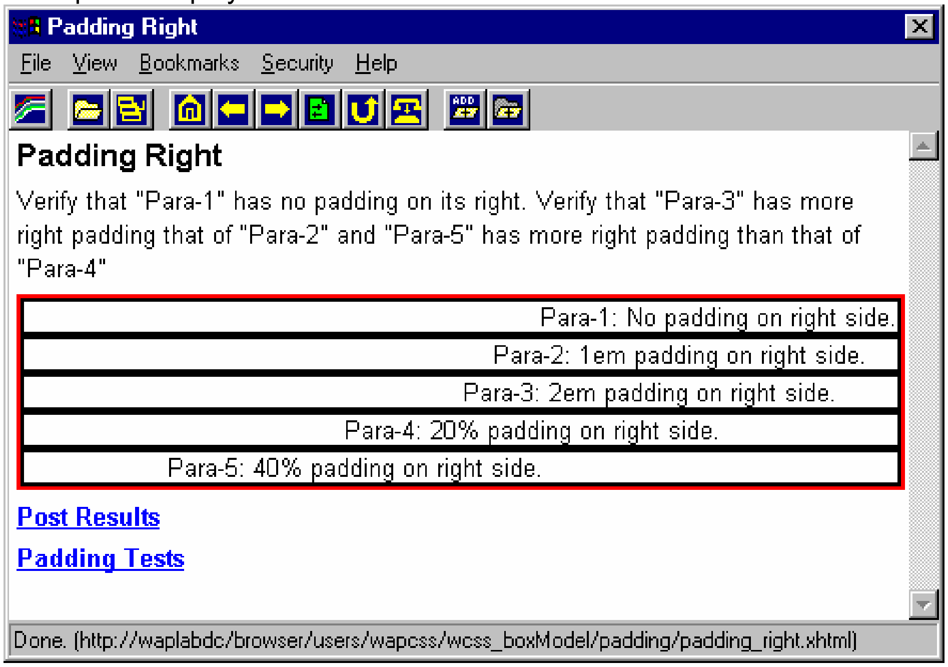
Assertion 2: When the PADDING-RIGHT property is set as a <percentage> value for an element in a style sheet, the user agent must apply the rule (shift the right padding) and the computed value for each instance of the element in the document tree.

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:



Padding Right

Verify that "Para-1" has no padding on its right. Verify that "Para-3" has more right padding that of "Para-2" and "Para-5" has more right padding than that of "Para-4"

Para-1: No padding on right side.

Para-2: 1em padding on right side.

Para-3: 2em padding on right side.

Para-4: 20% padding on right side.

Para-5: 40% padding on right side.

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_right.xhtml)

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.

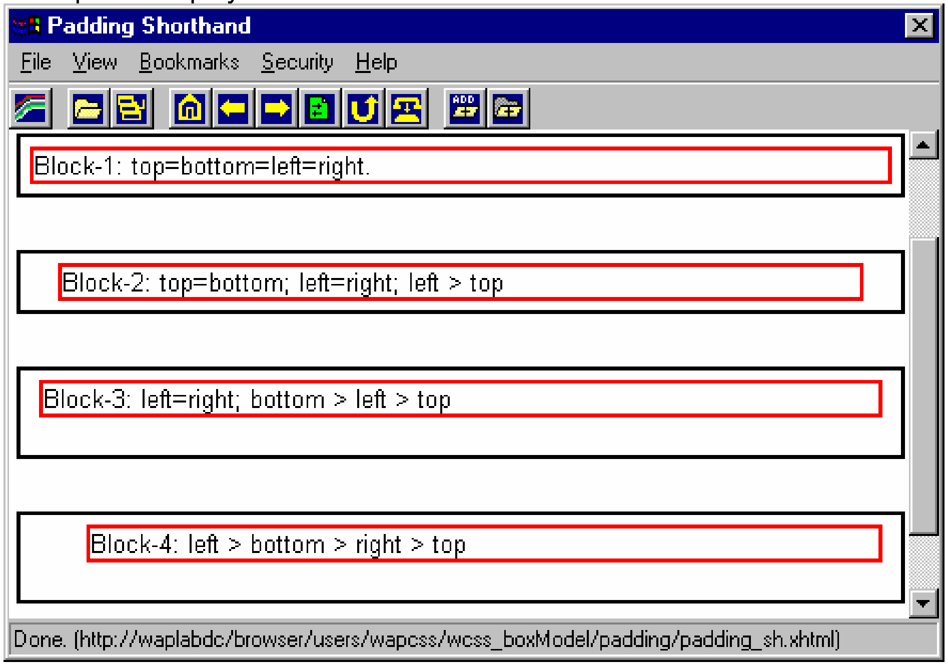
Traceability for all of above 4 assertions is:

[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 topside.
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:



Block-1: top=bottom=left=right.

Block-2: top=bottom; left=right; left > top

Block-3: left=right; bottom > left > top

Block-4: left > bottom > right > top

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/padding/padding_sh.xhtml)

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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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]	Section 10.3
[CSS2]	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-RIGTH-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".

45.19.19 **Border Width – Shorthand** **wcss_boxmodel/border_width/bwidth_sh.xhtml**

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-RIGTH-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]	Section 10.4
[CSS2]	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.

45.19.21 **Border Colors – Shorthand** **wcss_boxmodel/border_color/bcolors_sh.xhtml**

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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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 lines segments for each instance of the element in the document tree.

Assertion 3: When the BORDER-TOP-STYLE, BORDER-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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-RIGTH-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.

45.19.26 **Border Style – Shorthand 1** **wcss_boxmodel/border_style/bstyle_sh1.xhtml**

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.

45.19.27 **Border Style – Shorthand 2** **wcss_boxmodel/border_style/bstyle_sh2.xhtml**

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]	Section 10.5
[CSS2]	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.

45.19.28 **Border Style – Shorthand 3** **wcss_boxmodel/border_style/bstyle_sh3.xhtml**

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]	Section 10.5
[CSS2]	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.

45.19.31 Border Shorthand 1 wcss_boxmodel/border_shand/bshort_all1.xhtml

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.

45.19.36 **Named Colors 4** **wcss_boxmodel/border_shand/col_named4.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-3" are as described in the test.

9.7 More Box Model Tests:

45.19.37 Collapsing Margins **wcss_boxmodel/coll_margins.xhtml**

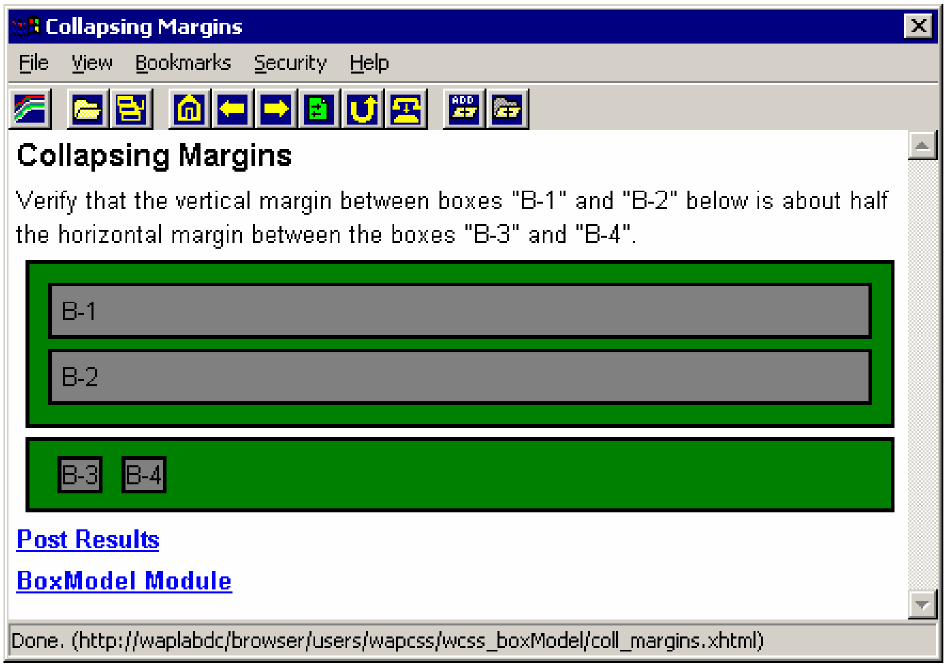
Assertion 1: When the margins are assigned to two are more adjoining boxes in the normal flow, then the user-agent must collapse the vertical margin between the adjoining boxes by taking the maximum of adjoining margins as the resulting combined margin.

Assertion 2: When the margins are assigned to two are 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".



Collapsing Margins

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

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/coll_margins.xhtml)

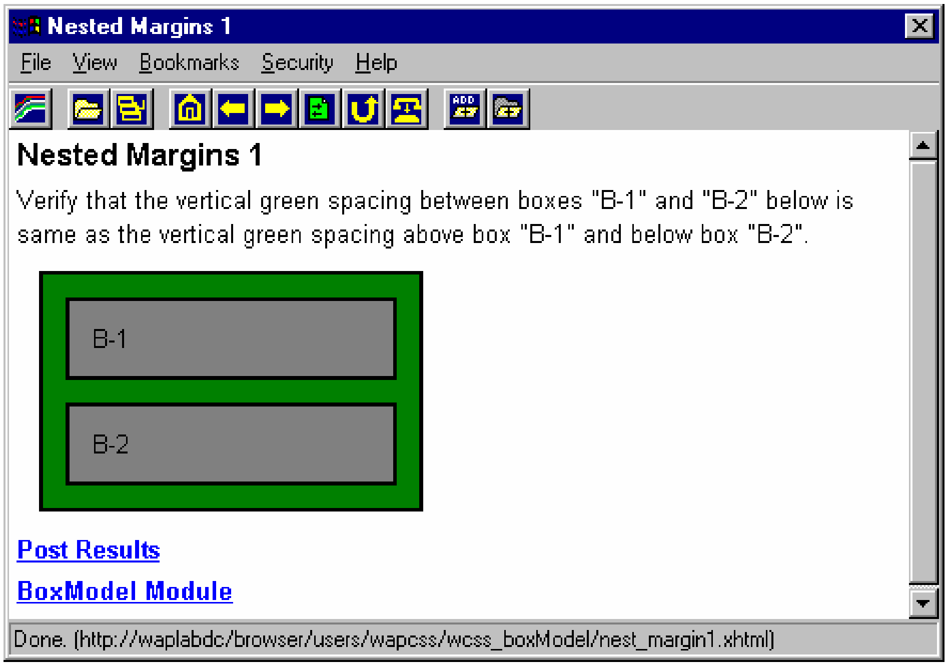
45.19.38 Nested Margins 1 wcss_boxmodel/nest_margin1.xhtml

Assertion: When the margins are assigned to two are 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.



Nested Margins 1

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

Post Results

BoxModel Module

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/nest_margin1.xhtml)

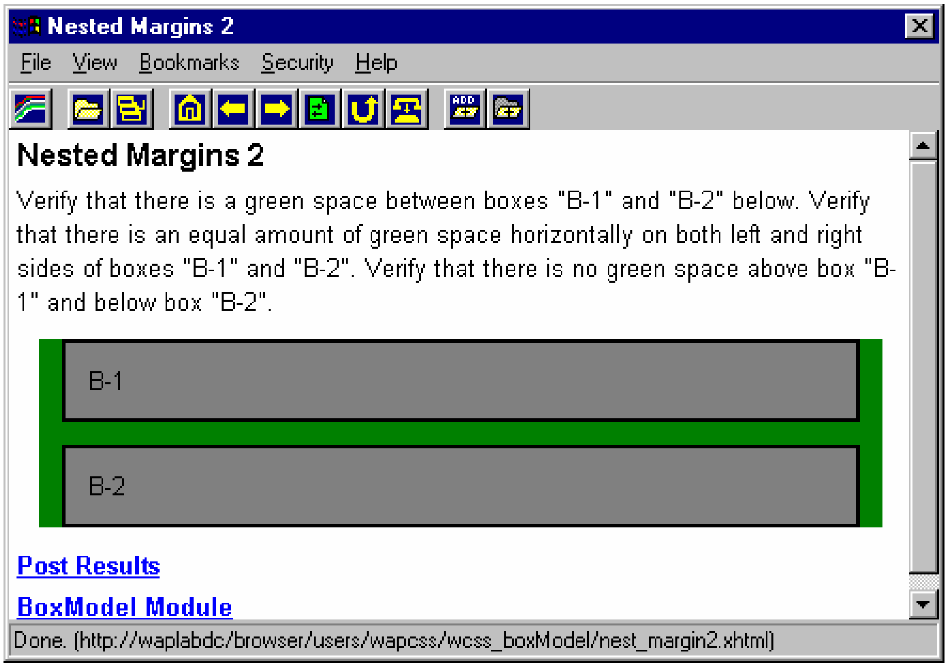
45.19.39 **Nested Margins 2** **wcss_boxmodel/nest_margin2.xhtml**

Assertion: When the margins are assigned to two are 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".



Nested Margins 2

Verify that there is a green space between boxes "B-1" and "B-2" below. 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".

Post Results

BoxModel Module

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/nest_margin2.xhtml)

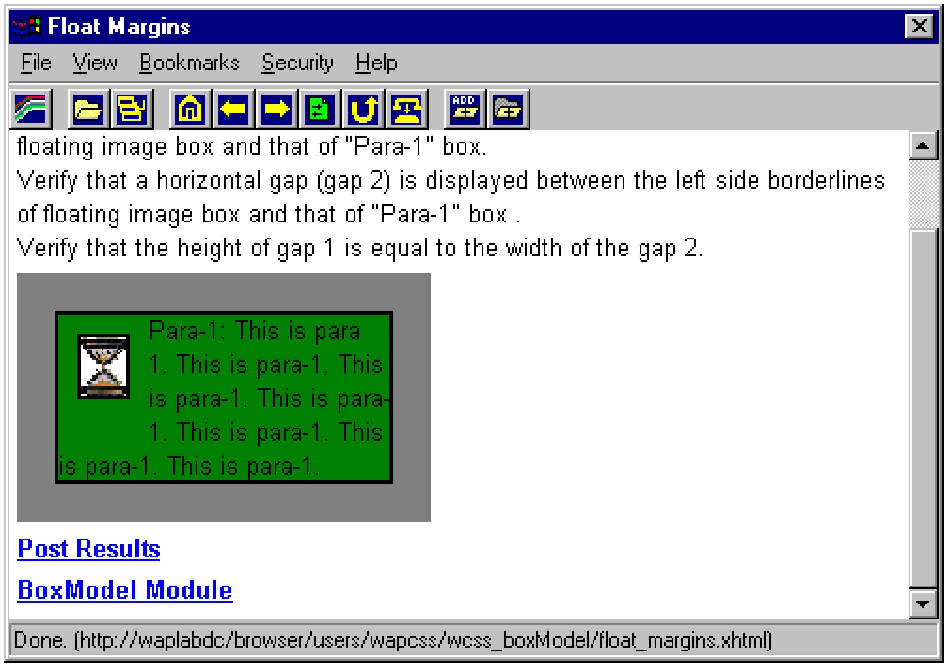
45.19.40 Float Margins wcss_boxmodel/float_margins.xhtml

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.



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.

Para-1: This is para 1. This is para-1. This is para-1. This is para-1. This is para-1. This is para-1. This is para-1.

Post Results

BoxModel Module

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/float_margins.xhtml)

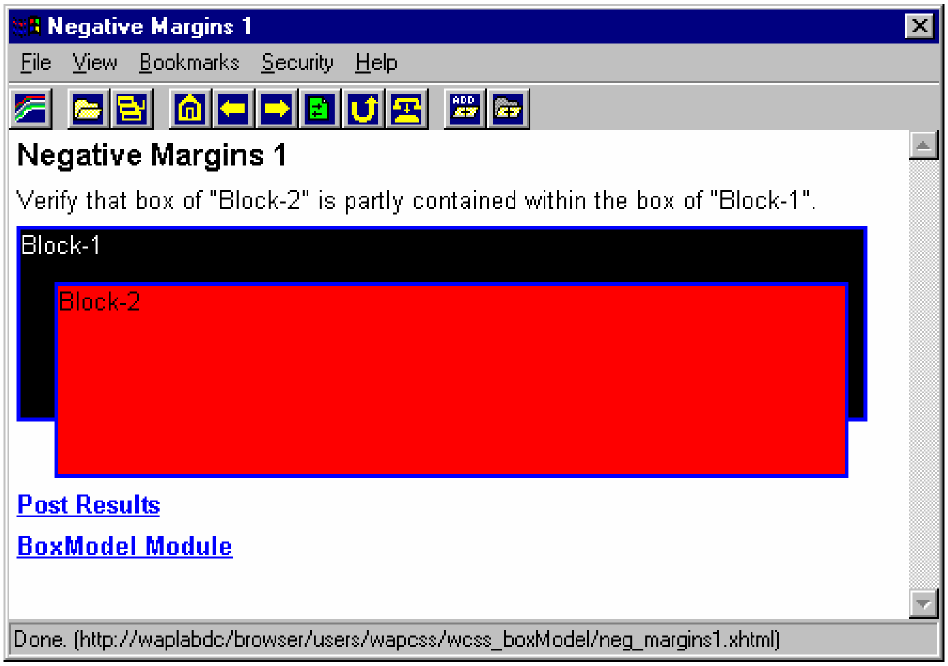
45.19.41 Negative Margins 1 **wcss_boxmodel/neg_margins1.xhtml**

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".



Negative Margins 1

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

Block-1

Block-2

Post Results

BoxModel Module

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/neg_margins1.xhtml)

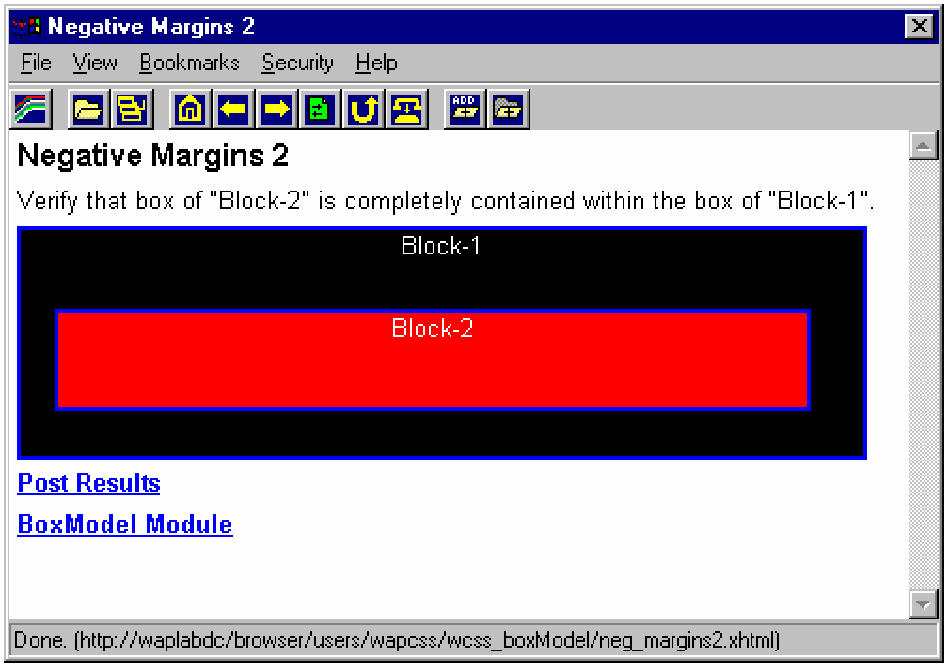
45.19.42 Negative Margins 2 wcss_boxmodel/neg_margins2.xhtml

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".



Negative Margins 2

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

Done. (http://waplabdc/browser/users/wapcss/wcss_boxModel/neg_margins2.xhtml)

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]	Section 11.3
[CSS2]	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]	Section 11.3
[CSS2]	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]	Section 11.3
[CSS2]	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]	Section 11.3
[CSS2]	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]	Section 11.3
[CSS2]	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.

45.20.33 Named Colors 3 wcss_colors/bg_shand/bgcol_named3.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-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]	Section 12.3
[CSS2]	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]	Section 12.4
[CSS2]	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]	Section 12.4
[CSS2]	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]	Section 12.5
[CSS2]	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]	Section 12.5
[CSS2]	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]	Section 12.5
[CSS2]	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 LISTS

12.1 LISTS

This section contains the tests for testing list properties (45.22.01 to 45.22.14). Each test in this section refers to SCR: WCSS-LISTS-C-001.

45.22.01 Discs **wcss_lists/ls_type/discs.xhtml**

Assertion 1: When the LIST-STYLE-TYPE property is set to 'disc' for an list-item element in a style sheet, the user agent must display a disc item in front of the element's textual content for each instance of the element in the document tree.

Assertion 2: When the LIST-STYLE-TYPE property is unset for an list-item element in a style sheet, the user agent must display a disc item in front of the element's textual content for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

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

45.22.02 Circles **wcss_lists/ls_type/circles.xhtml**

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.

Traceability:

[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…).

45.22.10 Marker Position wcss_lists/ls_position/list_style_position.xhtml

Assertion 1: When the LIST-STYLE-POSITION property is set to 'outside' for an list-item element in a style sheet, the user agent must display the marker box outside the principal block box for each instance of the element in the document tree.

Assertion 2: When the LIST-STYLE-POSITION property is unset for an list-item element in a style sheet, the user agent must display the marker box outside the principal block box for each instance of the element in the document tree.

Assertion 3: When the LIST-STYLE-POSITION property is set to 'inside' for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

Verification: Verify that the markers for list items of "List 1" and "List 2" are positioned outside the list item blocks and for "List 3", positioned inside. Example of outside and inside display of markers is shown below for information.

Eg: Outside marker placement:

- first item of List 1, marker placement, out side.
- second item of List 1, marker placement, out side.

Eg: Inside marker placement:

- first item of List 3, marker placement, in side.
- second item of List 3, marker placement, in side.

45.22.11 **Images as Markers** **wcss_lists/ls_image/list_style_image.xhtml**

Assertion 1: When the LIST-STYLE-IMAGE property is set to an URI for an list-item element in a style sheet, the user agent must use the defined image as the list-item marker for each instance of the element in the document tree.

Assertion 2: When the LIST-STYLE-IMAGE property is set to an URI for an list-item element in a style sheet, and if the defined image is unavailable, then the user agent must use the initial marker for each instance of the element in the document tree.

Assertion 3: When the LIST-STYLE-IMAGE property is unset for an list-item element in a style sheet, the user agent must use the initial marker value for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

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 discs are displayed as markers for list items of "List 1" and "List 2". Verify that image is displayed as markers for list items of "List 3". Note that it is OK, if the images are displayed in a truncated form.

45.22.12 **List Style 1** **wcss_lists/ls_shorthand/list_style1.xhtml**

Assertion 1: When the LIST-STYLE 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.

Assertion 2: When the LIST-STYLE property is set to 'inside' for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree.

Assertion 3: When the LIST-STYLE property is set to 'inside square' for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree; the marker must be a square item.

Traceability of above 3 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

Verification: Verify that "List 1" has circle items displayed as outside positioned markers. Verify that "List 2" has disc items displayed as inside positioned markers. Verify that "List 3" has square items displayed as inside positioned markers.

45.22.13 **List Style 2** **wcss_lists/ls_shorthand/list_style2.xhtml**

Assertion 1: When the LIST-STYLE property is set to an URI of an image for an list-item element in a style sheet, the user agent must use the defined image as the list-item marker in front of the element's textual content for each instance of the element in the document tree.

Assertion 2: When the LIST-STYLE property is set to 'inside' and a URI of an image for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree; the marker must be the defined image.

Traceability of above 3 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

Verification: Verify that "List 1" has images displayed as outside positioned markers. Verify that "List 2" has images displayed as inside positioned markers. Note that it is OK, if the images are displayed in a truncated form.

45.22.14 List Style 3 wcss_lists/ls_shorthand/list_style3.xhtml

Assertion 1: When the LIST-STYLE property is set to 'inside square' and a URI of an image for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree; the marker must be the defined image.

Assertion 2: When the LIST-STYLE property is set to 'inside square' and a URI of an image for an list-item element in a style sheet, the user agent must display the marker box as the first inline box in the principal block box for each instance of the element in the document tree; the marker must be square item if the defined image is unavailable.

Traceability of above 3 assertions:

[WAPCSS]	Section 13.2
[CSS2]	Section 12.6.2

Verification: Verify that "List 1" has images displayed as inside positioned markers. Verify that "List 2" has square items displayed as inside positioned markers. Note that it is OK, if the images are displayed in a truncated form.

13 TEXT

This module contains test cases for testing text properties, divided into five sub-modules (sections 14.1 to 14.13).

13.1 Indentation

This sub-module contains test cases for "text-indent" property. Start pages are described below (45.23.01 and 45.23.02). Each test in this section refers to SCR: WCSS-TEXT-INDENT-C-001.

45.23.01 Indent as Length wcss_text/indentation/ind_length.xhtml

Assertion 1: When the TEXT-INDENT property is set as a length value for an element in a style sheet, the user agent must display an indentation of the first line of the text in a block matching the defined value for each instance of the element in the document tree.

Assertion 2: When the TEXT-INDENT property is unset for an element in a style sheet, the user agent must not display any indentation of the first line of the text in a block for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 14.1
[CSS2]	Section 16.1

Verification: Verify that the first line of "Para 2" is indented farther than the first line of "Para 1". Verify that the first line of "Para 3" has no indentation at all and is displayed as a normal para.

45.23.02 Indent as Percent wcss_text/indentation/ind_percent.xhtml

Assertion: When the TEXT-INDENT property is set as a percentage value for an element in a style sheet, the user agent must display an indentation of the first line of the text in a block, corresponding to the percentage of the containing block width for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.1
[CSS2]	Section 16.1

Verification: Verify that the first line of "Para 2" is indented farther than the first line of "Para 1".

13.2 Alignment

This sub-module contains test cases for "text-align" property. Start pages are described below (45.23.03 and 45.23.04). Each test in this section refers to SCR: WCSS-TEXT-ALIGN-C-001.

45.23.03 Align L,R,C **wcss_text/alignment/align_lrc.xhtml**

Assertion 1: When the TEXT-ALIGN property is set to 'left' for an element in a style sheet, the user agent must align all the content text of the element to the left side of the padding area for each instance of the element in the document tree.

Assertion 2: When the TEXT-ALIGN property is set to 'right' for an element in a style sheet, the user agent must align all the content text of the element to the right side of the padding area for each instance of the element in the document tree.

Assertion 3: When the TEXT-ALIGN property is set to 'center' for an element in a style sheet, the user agent must align all the content text of the element to the center of the padding area for each instance of the element in the document tree.

Traceability of above 3 assertions:

[WAPCSS]	Section 14.2
[CSS2]	Section 16.2

Verification: Verify that the text in "Para 1" is left aligned, "Para 2" is right aligned and "Para 3" is center aligned.

45.23.04 Align Justify **wcss_text/alignment/align_just.xhtml**

Assertion: When the TEXT-ALIGN property is set to 'justify' for an element in a style sheet, the user agent must justify all the content text of the element for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.2
[CSS2]	Section 16.2

Verification: Verify that the text in "Para 1" is double justified. This means text is aligned on both left and right edges.

13.3 Decoration

This sub-module contains test cases for "text-decoration" property. Start pages are described below (45.23.05 to 45.23.07 & 45.23.14). Each test in this section refers to SCR: WCSS-TEXT-DEC-C-001.

45.23.05 No Decoration **wcss_text/decoration/dec_none.xhtml**

Assertion 1: When the TEXT-DECORATION property is unset for an element in a style sheet, the user agent must not produce any text decorations for each instance of the element in the document tree.

Assertion 2: When the TEXT-DECORATION property is set to 'none' for an element in a style sheet, the user agent must not produce any text decorations for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 14.3
[CSS2]	Section 16.3.1

Verification: Verify that the text of both "Para 1" and "Para 2" is displayed as normal text and have no special decorations like underline or blinking.

45.23.06 Underline **wcss_text/decoration/dec_uline.xhtml**

Assertion: When the TEXT-DECORATION property is set to 'underline' for an element in a style sheet, the user agent must underline all inline-level descendants of the element, excepted if the descendants have no content or no content text for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.3
[CSS2]	Section 16.3.1

Verification: Verify that the complete text in "Para 1" is displayed as underlined. Verify that the image in "Para 1" has no underline displayed. Verify that only the text "Boston" appears underlined in "Para 2".

45.23.07 **Blinking** **wcss_text/decoration/dec_blink.xhtml**

Assertion: When the TEXT-DECORATION property is set to 'blink' for an element in a style sheet, the user agent must make the text of all inline-level descendants of the element alternate between visible and invisible, excepted if the descendants have no content or no content text for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.3
[CSS2]	Section 16.3.1

Verification: Verify that the complete text in "Para 1" is displayed as blinking text. Verify that the image in "Para 1" does not appear as blinking image. Verify that only the word "Boston" appears with blinking text in "Para 2".

13.4 Transformation

This sub-module contains test cases for "text-transform" property. Start pages are described below (45.23.08 to 45.23.10). Each test in this section refers to SCR: WCSS-TEXT-TRANS-C-001.

45.23.08 Capitalization wcss_text/transformation/tr_cap.xhtml

Assertion 1: When the TEXT-TRANSFORMATION property is set to 'capitalize' for an element in a style sheet, the user agent must put the first character of each word of its text in uppercase for each instance of the element in the document tree.

Assertion 2: When the TEXT-TRANSFORMATION property is unset for an element in a style sheet, the user agent must not create any capitalization effects for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 14.4
[CSS2]	Section 16.5

Verification: Verify that the first character of each word in "Para 1" is displayed in uppercase. (Eg: "Many Happy Returns Of The Day"). Verify that the text in "Para 2" is displayed as normal text without any capitalization effects.

45.23.09 Uppercase wcss_text/transformation/tr_upper.xhtml

Assertion 1: When the TEXT-TRANSFORMATION property is set to 'uppercase' for an element in a style sheet, the user agent must put all characters of each word of its text in uppercase for each instance of the element in the document tree.

Assertion 2: When the TEXT-TRANSFORMATION property is set to 'none' for an element in a style sheet, the user agent must not create any capitalization effects for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 14.4
[CSS2]	Section 16.5

Verification: Verify that the entire text of "Para 1" is displayed in uppercase characters only. Verify that the text in "Para 2" is displayed as normal text without any capitalization effects.

45.23.10 **Lowercase** **wcss_text/transformation/tr_lower.xhtml**

Assertion: When the TEXT-TRANSFORMATION property is set to 'lowercase' for an element in a style sheet, the user agent must put all characters of each word of its text in lowercase for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.4
[CSS2]	Section 16.5

Verification: Verify that the entire text of "Para 1" is displayed in lowercase characters only.

13.5 Whitespace

This sub-module contains test cases for "whitespace" property. Start pages are described below (45.23.11 to 45.23.13). Each test in this section refers to SCR: WCSS-TEXT-WS-C-001.

45.23.11 Normal Space **wcss_text/whitespace/ws_normal.xhtml**

Assertion 1: When the WHITE-SPACE property is set to 'normal' for an element in a style sheet, the user agent must collapse sequences of white space, and break lines as necessary to fill line boxes for each instance of the element in the document tree.

Assertion 2: When the WHITE-SPACE property is unset for an element in a style sheet, the user agent must collapse sequences of white space, and break lines as necessary to fill line boxes for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 14.5
[CSS2]	Section 16.6

Verification: Verify that the text in both the "Para 1" and "Para 2" is displayed normally with lines wrapped and intra-word white space collapsed into single space. Verify that in both the paragraphs, in addition to other line breaks, lines always break immediately after '*' signs (there is a "br" element immediately after '*' character in the source code).

45.23.12 Preserve Space **wcss_text/whitespace/ws_pre.xhtml**

Assertion: When the WHITE-SPACE property is set to 'pre' for an element in a style sheet, the user agent must not collapse sequences of white space and must break the lines only at new lines in the source or at occurrences of 'br' tag for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.5
[CSS2]	Section 16.6

Verification: Verify that the text in "Para 1" is displayed with large intra-word white space preserved. Verify that line breaks occur only after '*' signs and '#' signs (there is a "br" element immediately after every '*' character and new-line after every '#' character in the source code).

45.23.13 No Wrap **wcss_text/whitespace/ws_nowrap.xhtml**

Assertion: When the WHITE-SPACE property is set to 'nowrap' for an element in a style sheet, the user agent must collapse sequences of white space but suppress line breaks within the text except for those created by the 'br' tag for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 14.5
[CSS2]	Section 16.6

Verification: Verify that the text in "Para 1" is displayed with large intra-word white space reduced to single space. Verify that lines wrap immediately after '*' characters only and not anywhere else. (There is a "br" element immediately after every '*' character and new-line after every '#' character in the source code.)

13.6 More Text tests

45.23.14 Blink Uline wcss_text/decoration/dec_blink_uline.xhtml

Assertion 1: When the TEXT-DECORATION property is set to 'underline' for an element in a style sheet, the user agent must underline all inline-level descendants of the element, excepted if the descendants have no content or no content text for each instance of the element in the document tree.

Assertion 2: When the TEXT-DECORATION property is set to 'blink' for an element in a style sheet, the user agent must make the text of all inline-level descendants of the element alternate between visible and invisible, excepted if the descendants have no content or no content text for each instance of the element in the document tree.

Traceability of above two assertions:

[WAPCSS]	Section 14.3
[CSS2]	Section 16.3.1

Assertion 3: When a CSS property is defined with two or more values separated by || syntax, then one or more of the || separated options can be assigned as values to the property in any order.

Traceability:

[WAPCSS]	Section 6.1
[CSS2]	Section 1.3.2

Verification: Verify that both of the lines “Line-1” and “Line-2” are displayed blinking and underlined.

14 VISUAL EFFECTS

This module contains the tests for 'visibility' property. Starting pages are described below (45.24.01 to 45.24.03). Each test in this section refers to SCR: WCSS-VEFFECT-C-001.

45.24.01 Visible and Hidden **wcss_vEffects/vis_hidden.xhtml**

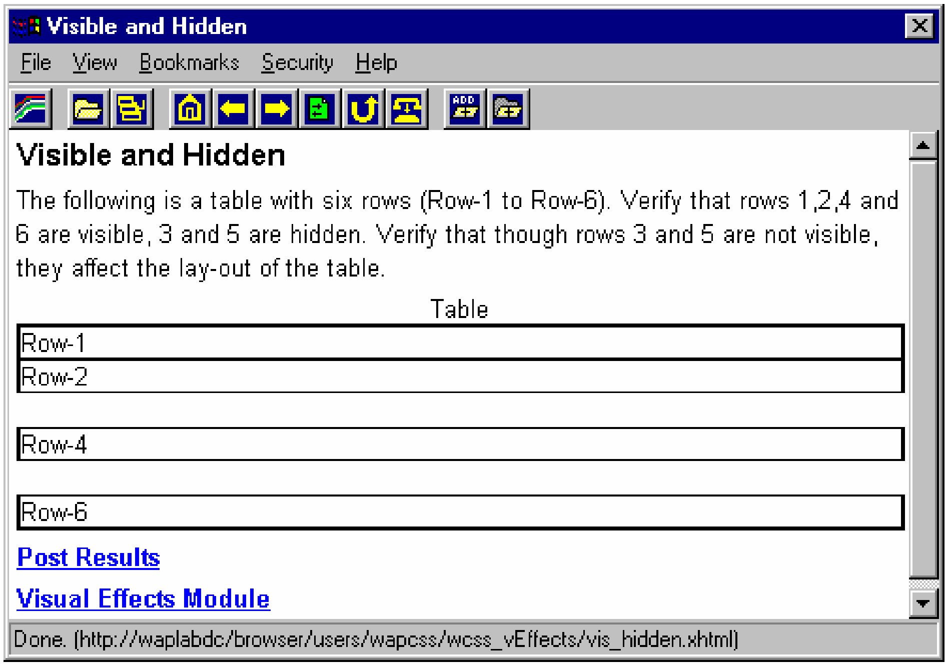
Assertion 1: When the VISIBILITY property is set to 'visible' for an element in a style sheet, the user agent must generate a visible element's box for each instance of the element in the document tree

Assertion 2: When the VISIBILITY property is set to 'hidden' for an element in a style sheet, the user agent must generate an invisible (fully transparent) element's box, but still affecting layout for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 15
[CSS2]	Section 11.2

Verification: Verify that in the six-row table, only "Row-1", "Row-2","Row-4" and "Row-6" are visible. Verify that "Row-3" and "Row-5" are not visible, but still affect the layout of the table. This means "Row-2" and "Row-4" don't stick together. Similarly "Row-4" and "Row-6" don't stick together.



Visible and Hidden

The following is a table with six rows (Row-1 to Row-6). Verify that rows 1,2,4 and 6 are visible, 3 and 5 are hidden. Verify that though rows 3 and 5 are not visible, they affect the lay-out of the table.

Table

Row-1
Row-2

Row-4

Row-6

Post Results

Visual Effects Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vEffects/vis_hidden.xhtml)

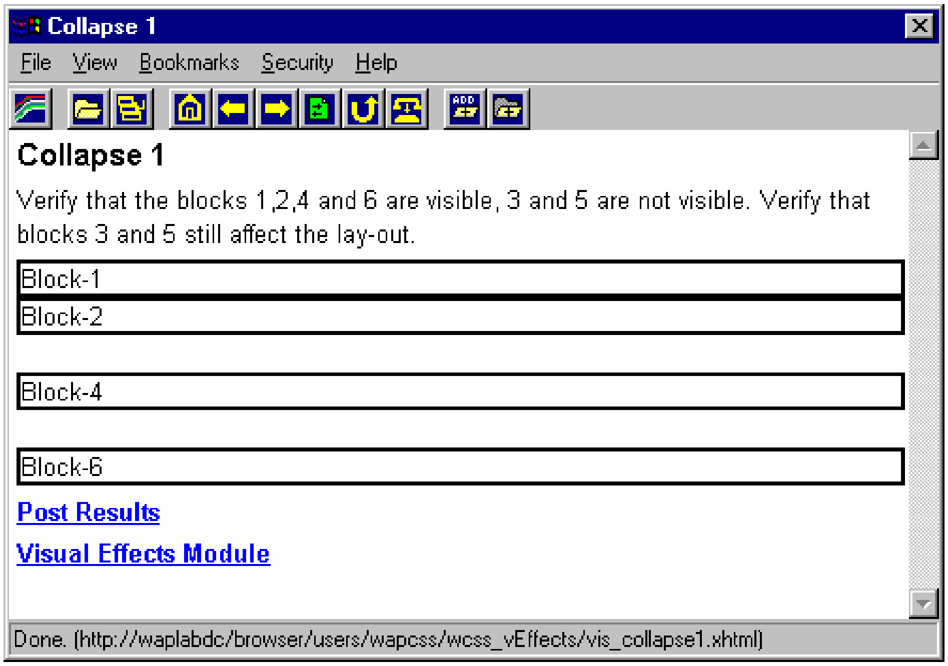
45.24.02 Collapse 1 wcss_vEffects/vis_collapse1.xhtml

Assertion: When the VISIBILITY property is set to 'collapse' for an element, other than rows or columns, in a style sheet, the user agent must generate an invisible (fully transparent) element's box, but still affecting layout for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 15
[CSS2]	Section 11.2

Verification: Verify that only "Block-1", "Block-2","Block-4" and "Block-6" are visible. Verify that "Block-3" and "Block-5" are not visible, but still affect the layout of the other blocks. This means "Block-2" and "Block-4" don't stick together. Similarly "Block-4" and "Block-6" don't stick together.



Collapse 1

Verify that the blocks 1,2,4 and 6 are visible, 3 and 5 are not visible. Verify that blocks 3 and 5 still affect the lay-out.

Block-1

Block-2

Block-4

Block-6

Post Results

Visual Effects Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vEffects/vis_collapse1.xhtml)

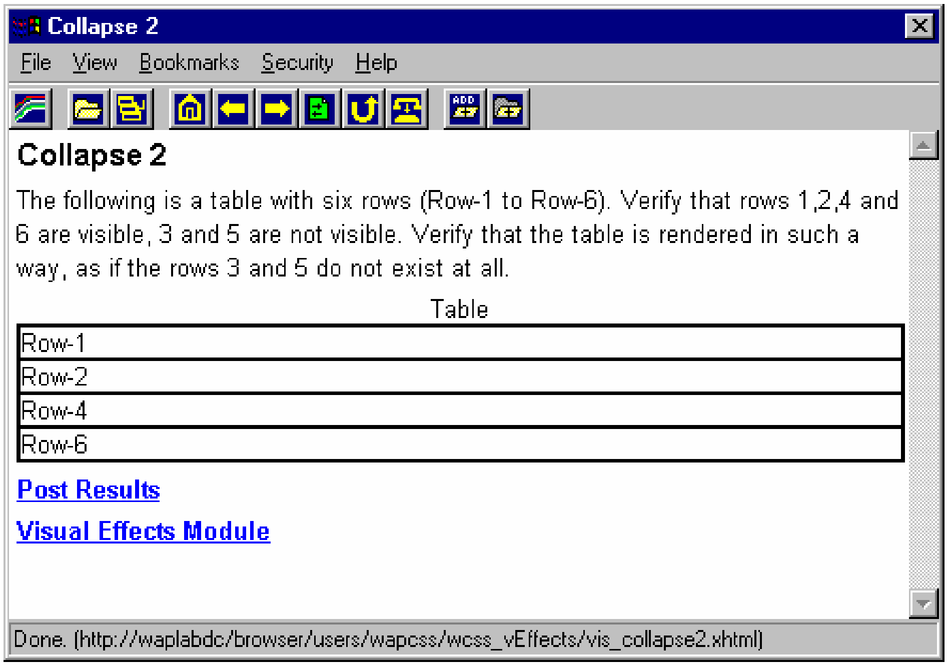
45.24.03 **Collapse 2** **wcss_vEffects/vis_collapse2.xhtml**

Assertion: When the VISIBILITY property is set to 'collapse' for the row or column element, in a style sheet, the user agent must remove the row(s) or the column(s) from the layout and so forth reorder dynamically the display of the table for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 15
[CSS2]	Section 11.2

Verification: Verify that in the six-row table, only "Row-1", "Row-2","Row-4" and "Row-6" are visible. Verify that "Row-3" and "Row-5" are not visible and don't affect the layout of the table. This means the table is rendered in such a way that "Row-3" and "Row-5" does not exist at all.



Collapse 2

The following is a table with six rows (Row-1 to Row-6). Verify that rows 1,2,4 and 6 are visible, 3 and 5 are not visible. Verify that the table is rendered in such a way, as if the rows 3 and 5 do not exist at all.

Table

Row-1
Row-2
Row-4
Row-6

Post Results

Visual Effects Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vEffects/vis_collapse2.xhtml)

15 VISUAL FORMATTING

This module contains tests for testing visual formatting properties (display, float, clear, height, width and vertical-align properties).

15.1 Display Properties

This sub-module contains test cases for "display" property. Start pages are described below (45.25.01 to 45.25.04 and 45.25.22). Each test in this section refers to SCR: WCSS-VFORM-DISP-C-001.

45.25.01 Block Display **wcss_vFormat/display/block_display.xhtml**

Assertion: When the DISPLAY property is set to 'block' for an element in a style sheet, the user agent must generate a principal block box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.1
[CSS2]	Section 9.2.5

Verification: Verify that the text "B1", "B2" and "B3" appears as three separate blocks (or as separate lines), but not as a single line.

45.25.02 Inline Display **wcss_vFormat/display/inl_display.xhtml**

Assertion: When the DISPLAY property is set to 'inline' for an element in a style sheet, the user agent must generate one or more inline boxes for each instance of the element in the document tree

Traceability:

[WAPCSS]	Section 16.1
[CSS2]	Section 9.2.5

Verification: Verify that the text "B1", "B2" and "B3" appears as a single line, but not as separate blocks.

45.25.03 **No Display** **wcss_vFormat/display/no_display.xhtml**

Assertion: When the DISPLAY property is set to 'none' for an element in a style sheet, the user agent must generate no boxes in the formatting structure for each instance of the element in the document tree and its descendants.

Traceability:

[WAPCSS]	Section 16.1
[CSS2]	Section 9.2.5

Verification: Verify that in "Para-1", the text "ABCD" appears as "AD". Verify that the text "BC" in-between is not displayed.

45.25.04 **List-Item Display** **wcss_vFormat/display/li_display.xhtml**

Assertion: When the DISPLAY property is set to 'list-item' for an element in a style sheet, the user agent must generate a principal block box and a list-item inline box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.1
[CSS2]	Section 9.2.5

Verification: Verify that the names of the fruits appear as a bulleted list.

15.2 Float Positioning

This sub-section contains test cases for "float" property. Start pages are described below (45.25.05 to 45.25.07 and 45.25.23 to 45.25.25). Each test in this section refers to SCR: WCSS-VFORM-FLOAT-C-001.

45.25.05 Float Left **wcss_vFormat/float/float_left.xhtml**

Assertion 1: When the FLOAT property is set to 'left' for an element in a style sheet, the user agent must generate a block box that is floated to the left and make the content flown on the right side of the box, starting at the top for each instance of the element in the document tree.

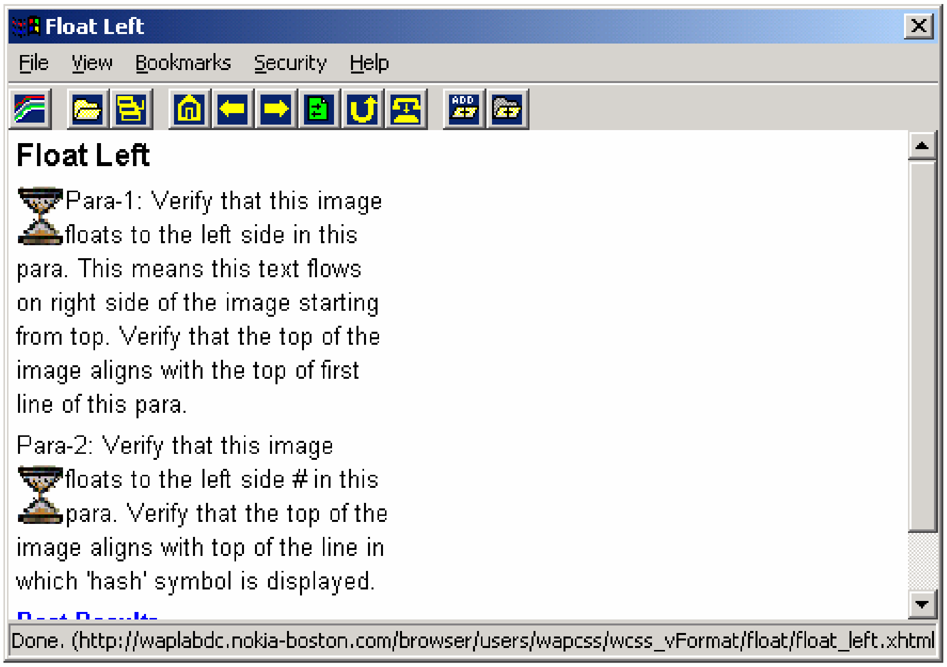
Assertion 2: The top of the floated box must be aligned with the top of the current line box.

Traceability:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that in "Para-1", the image floats to left of the para and the text flows along the right side of the image starting from top. This means text is not displayed on the left side of the image. Verify that the top of the image aligns with the top of first line of this para.

Verify that in “Para-2” also an image floats to the left side. Verify that the top of this image aligns with top of the line in which 'hash' symbol is displayed.



Float Left

Para-1: Verify that this image floats to the left side in this para. This means this text flows on right side of the image starting from top. Verify that the top of the image aligns with the top of first line of this para.

Para-2: Verify that this image floats to the left side # in this para. Verify that the top of the image aligns with top of the line in which 'hash' symbol is displayed.

Post Results

Done. (http://waplabdc.nokia-boston.com/browser/users/wapcss/wcss_vFormat/float/float_left.xhtml

45.25.06 **Float Right** **wcss_vFormat/float/float_right.xhtml**

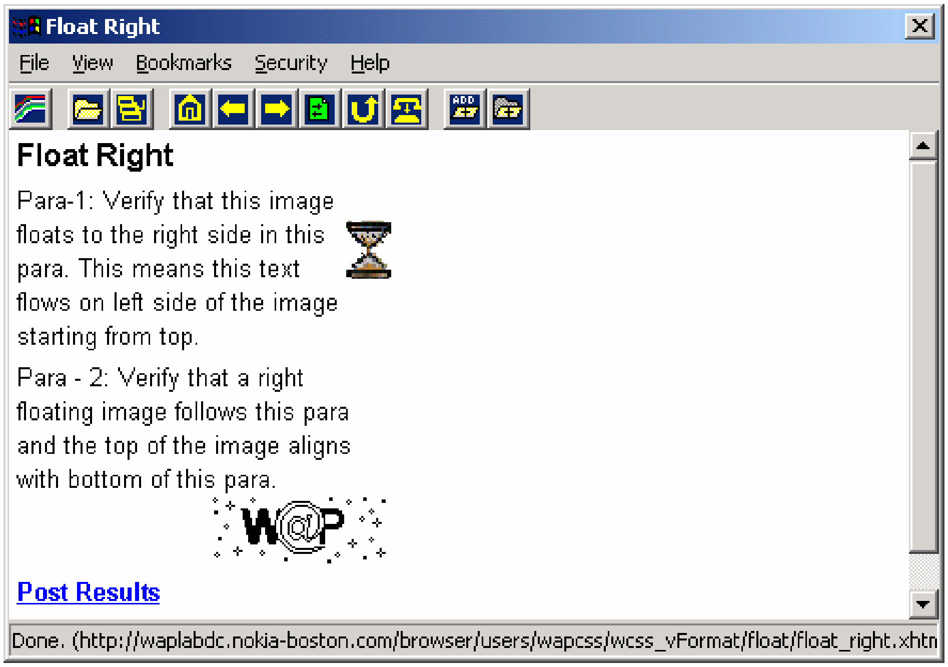
Assertion: When the FLOAT property is set to 'right' for an element in a style sheet, the user agent must generate a block box that is floated to the right and make the content flown on the left side of the box, starting at the top for each instance of the element in the document tree.

Assertion 2: The top of the floated box must be aligned with the bottom of the preceding block box if no line box exists.

Traceability:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that in "Para-1", the image floats to right of the para and the text flows along the left side of the image starting from top. This means text is not displayed on the right side of the image. Verify that a right floating image follows the “Para-2” and the top of the image is aligned with the bottom of the “Para-2”.



Float Right

Para-1: Verify that this image floats to the right side in this para. This means this text flows on left side of the image starting from top.

Para - 2: Verify that a right floating image follows this para and the top of the image aligns with bottom of this para.

Post Results

Done. (http://waplabdc.nokia-boston.com/browser/users/wapcss/wcss_vFormat/float/float_right.xhtn

45.25.07 **Float None** **wcss_vFormat/float/float_none.xhtml**

Assertion 1: When the FLOAT property is set to 'none' for an element in a style sheet, the user agent must not make the box floated for each instance of the element in the document tree.

Assertion 2: When the FLOAT property is unset for an element in a style sheet, the user agent must not make the box floated for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that both in "Para-1" and "Para-2", image does not float to either side of the para, and is positioned in-between two pound signs ("#").

15.3 Float Flow Control

This sub-module contains test cases for "clear" property. Start pages are described below (45.25.08 to 45.25.12 and 45.25.26 to 45.25.27). Each test in this section refers to SCR: WCSS-VFORM-CLEAR-C-001.

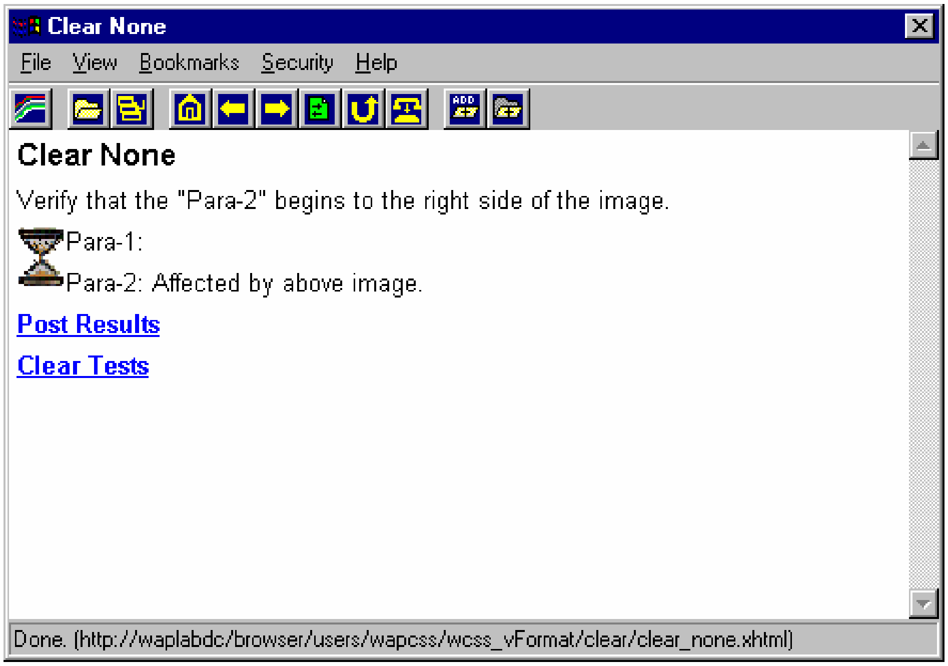
45.25.08 Clear None wcss_vFormat/clear/clear_none.xhtml

Assertion: When the CLEAR property is set to 'none' for an element in a style sheet, the user agent must not add any constraints on the box's position with respect to floats for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the position of left floating image of "Para-1" affects the display of "Para-2". This means that "Para-2" begins at the right side of image.



Clear None

Verify that the "Para-2" begins to the right side of the image.

Para-1:

Para-2: Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_none.xhtml)

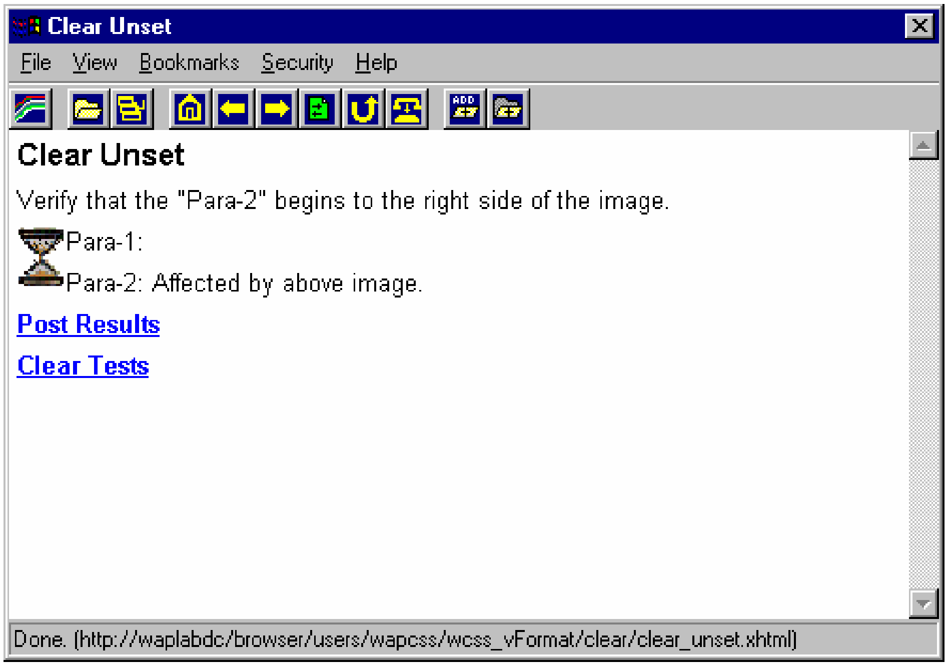
45.25.09 Clear Unset wcss_vFormat/clear/clear_unset.xhtml

Assertion: When the CLEAR property is unset for an element in a style sheet, the user agent must not add any constraints on the box's position with respect to floats for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the position of left floating image of "Para-1" affects the display of "Para-2". This means that "Para-2" begins at the right side of image.



Clear Unset

Verify that the "Para-2" begins to the right side of the image.

Para-1:

Para-2: Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_unset.xhtml)

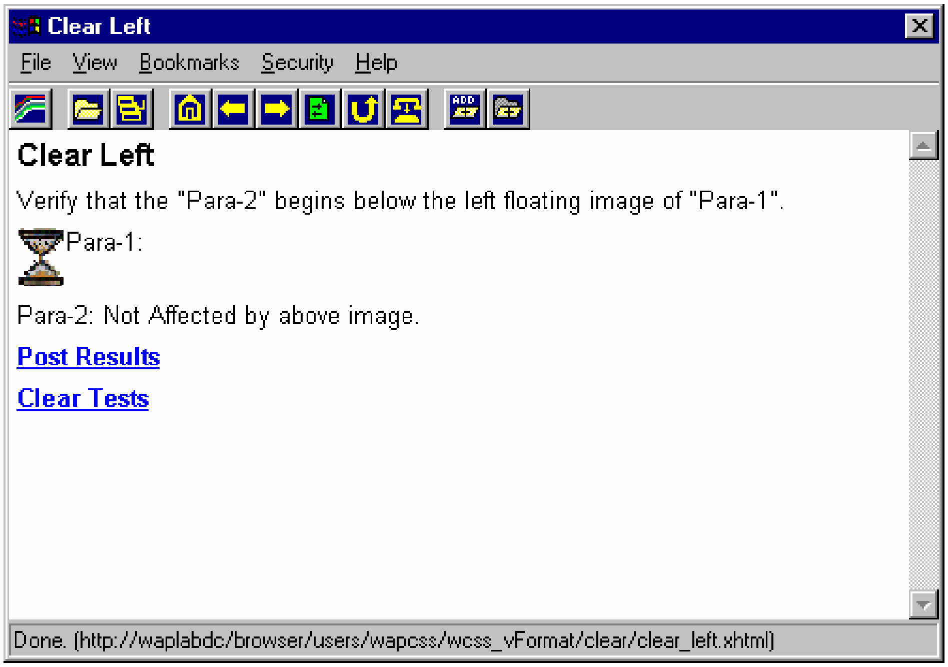
45.25.10 Clear Left wcss_vFormat/clear/clear_left.xhtml

Assertion: When the CLEAR property is set to 'left' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any left-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the left floating image of “Para-1” does not affect the display of “Para-2”. This means that "Para-2" must begin below the image.



Clear Left

Verify that the "Para-2" begins below the left floating image of "Para-1".

Para-1:

Para-2: Not Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_left.xhtml)

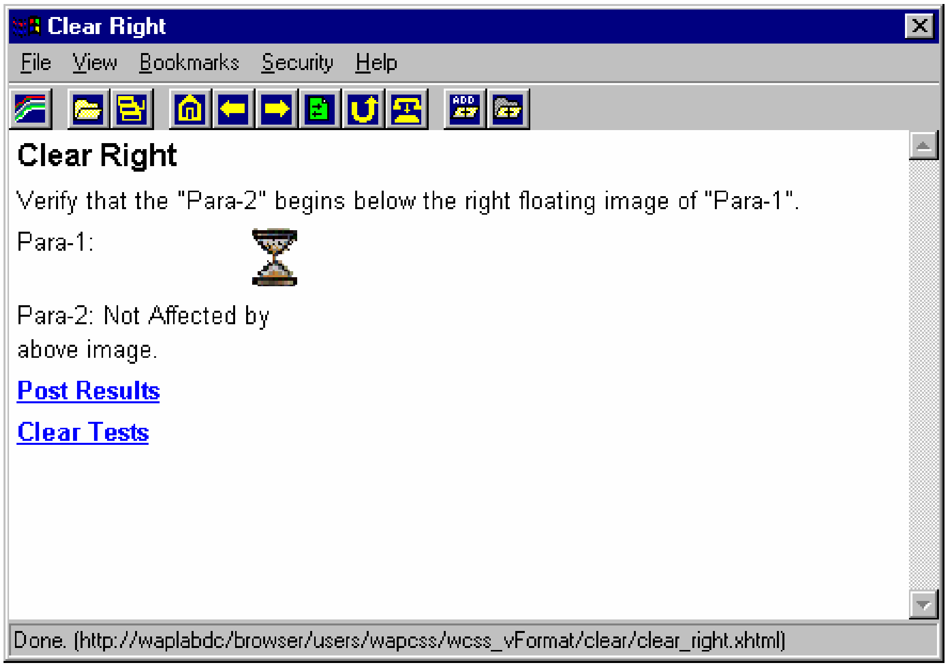
45.25.11 Clear Right **wcss_vFormat/clear/clear_right.xhtml**

Assertion: When the CLEAR property is set to 'right' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any right-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the right floating image of “Para-1” does not affect the display of “Para-2”. This means the "Para-2" must begin below the image.



Clear Right

Verify that the "Para-2" begins below the right floating image of "Para-1".

Para-1:

Para-2: Not Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_right.xhtml)

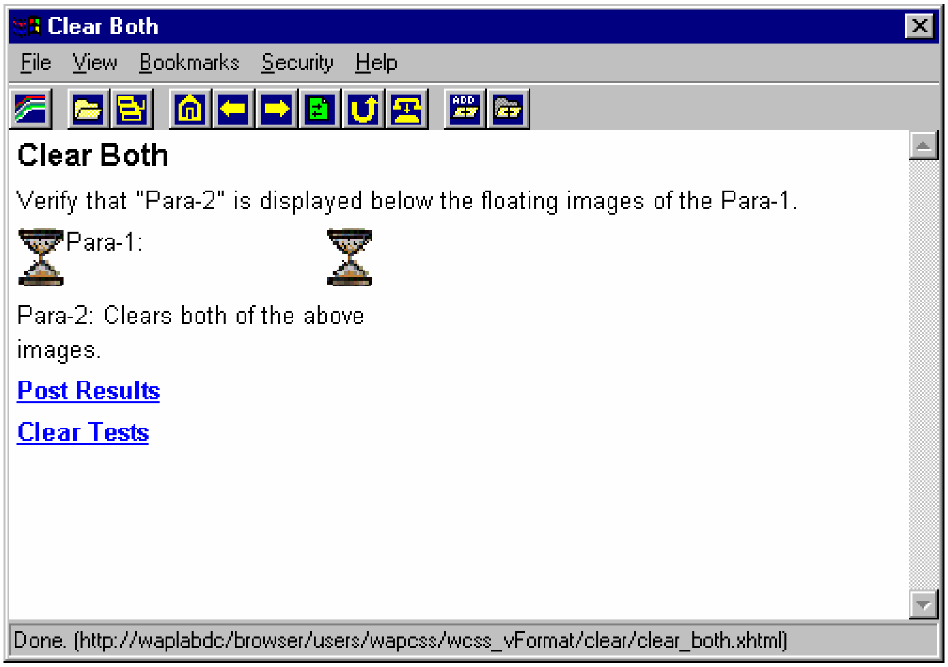
45.25.12 Clear Both wcss_vFormat/clear/clear_both.xhtml

Assertion: When the CLEAR property is set to 'both' for an element in a style sheet, the user agent must display the box below the bottom of all floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the display of "Para-2" is not affected by either of the left or right side floating images of "Para-1. This means that "Para-2" begins after clearing both of the right side and left-side floating images.



Clear Both

Verify that "Para-2" is displayed below the floating images of the Para-1.

Para-1:

Para-2: Clears both of the above images.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_both.xhtml)

15.4 Content Width & Height

This sub-module contains test cases for "width", "height" and "vertical-align" properties. Start pages are described below (45.25.13 to 45.25.21). Each test in this section refers to SCR: WCSS-VFORM-SIZE-C-001.

45.25.13 Size-Length **wcss_vFormat/dimensions/size_length.xhtml**

Assertion 1: When the WIDTH property is set as a length value for an element in a style sheet, the user agent must display the width of the element's box equal to the defined value for each instance of the element in the document tree.

Assertion 2: When the HEIGHT property is set as a length value for an element in a style sheet, the user agent must display the height of the element's box equal to the defined value for each instance of the element in the document tree.

Traceability of above 2 assertions:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.2

Verification: Verify that height and width of "Box-2" are more than that of "Box-1".

45.25.14 Size-Percent **wcss_vFormat/dimensions/size_percent.xhtml**

Assertion 1: When the WIDTH property is set as a percentage value for a element in a style sheet, the user agent must display the width of the element's box equal to the defined percentage value for each instance of the element in the document tree; the percentage is calculated with respect to the width of the generated box's containing block.

Assertion 2: When the HEIGHT property is set as a percentage value for a element in a style sheet, the user agent must display the height of the element's box equal to the defined percentage value for each instance of the element in the document tree; the percentage is calculated with respect to the height of the generated box's containing block

Traceability of above 2 assertions:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.2

Verification: Verify that height and width of "Box-2" are more than that of "Box-1"

45.25.15 VAlign Unset wcss_vFormat/dimensions/valign_unset.xhtml

Assertion: When the VERTICAL-ALIGN property is unset for an inline element in a style sheet, the user agent must align the baseline of the box with the baseline of the parent box, or, if the box does not have a baseline, align the bottom of the box with the parent's baseline for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the baseline of the text "child" box is aligned with the baseline of the text "parent" (not with the lowest point of letter 'p' of parent).

45.25.16 VAlign Baseline wcss_vFormat/dimensions/valign_baseline.xhtml

Assertion: When the VERTICAL-ALIGN property is set to 'baseline' for an inline element in a style sheet, the user agent must align the baseline of the box with the baseline of the parent box, or, if the box does not have a baseline, align the bottom of the box with the parent's baseline for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the baseline of the text "child" box is aligned with the baseline of the text "parent" (not with the lowest point of letter 'p' of parent).

45.25.17 VAlign Sub wcss_vFormat/dimensions/valign_sub.xhtml

Assertion: When the VERTICAL-ALIGN property is set to 'sub' for an inline element in a style sheet, the user agent must lower the baseline of the box to the proper position for subscripts of the parent's box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the "n" is displayed as sub-script of "X". (Eg: X_n)

45.25.18 VAlign Super **wcss_vFormat/dimensions/valign_super.xhtml**

Assertion: When the VERTICAL-ALIGN property is set to 'super' for an inline element in a style sheet, the user agent must raise the baseline of the box to the proper position for subscripts of the parent's box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the "2" is displayed as super-script of "X" such as in mathematical representation of square of X.

45.25.19 VAlign Top **wcss_vFormat/dimensions/valign_top.xhtml**

Assertion: When the VERTICAL-ALIGN property is set to 'top' for an inline element in a style sheet, the user agent must align the top of the box with the top of the line box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the top of the text of "child" is aligned with the top line of the image displayed in the line.

45.25.20 VAlign Middle **wcss_vFormat/dimensions/valign_middle.xhtml**

Assertion: When the VERTICAL-ALIGN property is set to 'middle' for an inline element in a style sheet, the user agent must align the vertical midpoint of the box with the baseline of the parent box plus half the x-height of the parent for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the vertical mid-point of the image is aligned with the baseline of the "Parent" text plus half the x-height of parent's font.

45.25.21 VAlign Bottom **wcss_vFormat/dimensions/valign_bottom.xhtml**

Assertion: When the VERTICAL-ALIGN property is set to 'bottom' for an inline element in a style sheet, the user agent must align the bottom of the box with the bottom of the line box for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.4
[CSS2]	Section 10.8.1

Verification: Verify that the bottom of the text of "q-child" is aligned with the bottom line of the image.

15.5 More Formatting tests

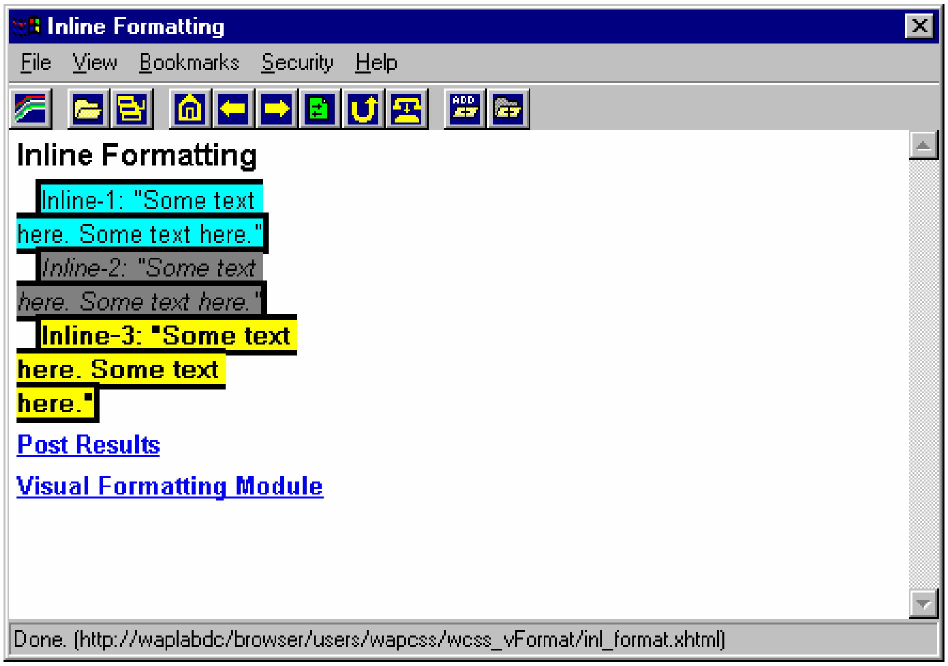
45.25.22 Inline formatting **wcss_vFormat/inl_format.xhtml**

Assertion: When a line box contains number of inline boxes, then UA must render them laid out horizontally one after another respecting the horizontal margins, borders and padding between these boxes.

Traceability:

[WAPCSS]	Section 16.1
[CSS2]	Section 9.4.2

Verification: Verify that inline blocks "Inline-1" to "Inline-3" are displayed with solid borders and are laid out horizontally one after another, flowing into multiple lines. Verify that when any of these blocks is split into more than one line, no vertical borderline is displayed at the points of splitting.



Inline Formatting

Inline-1: "Some text here. Some text here."

Inline-2: "Some text here. Some text here."

Inline-3: "Some text here. Some text here."

Post Results

Visual Formatting Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/inl_format.xhtml)

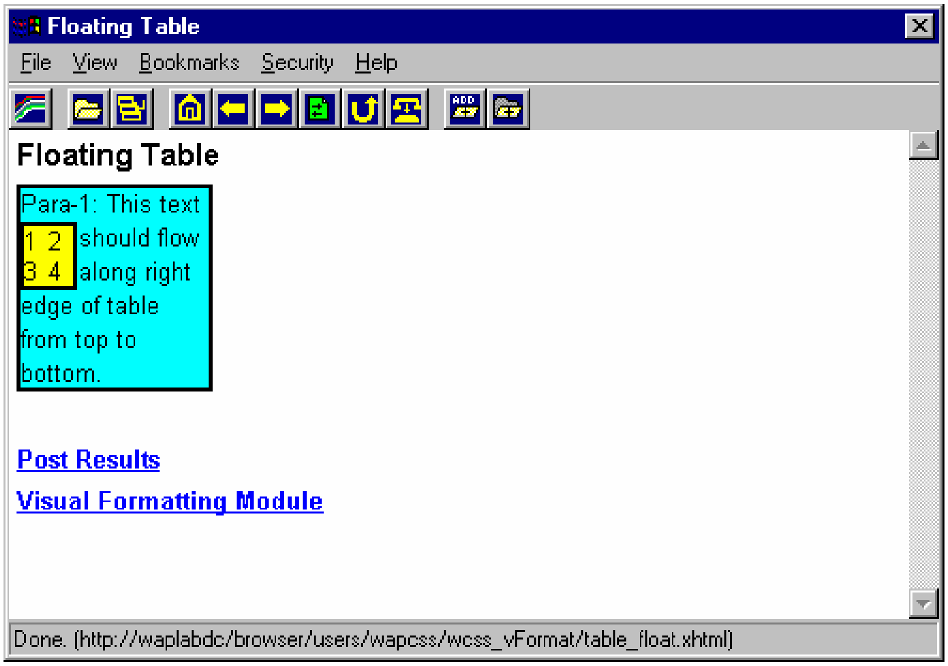
45.25.23 Floating Table **wcss_vFormat/table_float.xhtml**

Assertion: When the FLOAT property is set to 'left' for an element in a style sheet, the user agent must generate a block box that is floated to the left and make the content flown on the right side of the box, starting at the top for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that in "Para-1", a table containing numbers "1" to "4" floats to the left of the para and the text of the para flows along the right side of the table starting from top. This means text is not displayed on the left side of the table.



Floating Table

Para-1: This text should flow along right edge of table from top to bottom.

Post Results

Visual Formatting Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/table_float.xhtml)

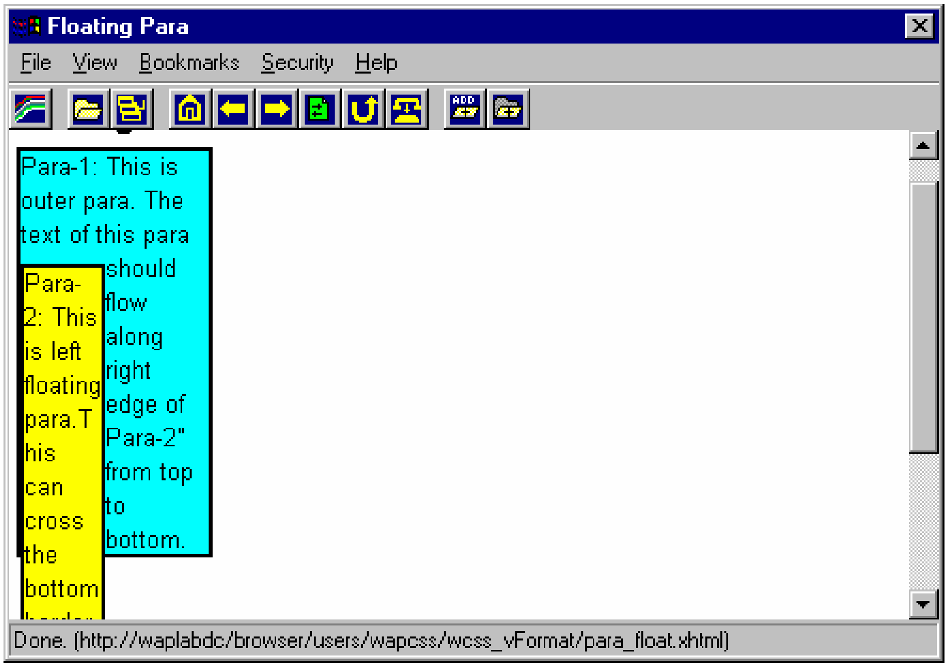
45.25.24 Floating Para wcss_vFormat/para_float.xhtml

Assertion: When the FLOAT property is set to 'left' for an element in a style sheet, the user agent must generate a block box that is floated to the left and make the content flown on the right side of the box, starting at the top for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that "Para-2" is displayed floating left within the block box of "Para-1". Verify that the text of "Para-1" flows top to bottom on the right side of "Para-2". *Note that "Para-2" can extend beyond the bottom borderline of "Para-1".*



Para-1: This is outer para. The text of this para should flow along right edge of Para-2" from top to bottom.

Para-2: This is left floating para.This can cross the bottom

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/para_float.xhtml)

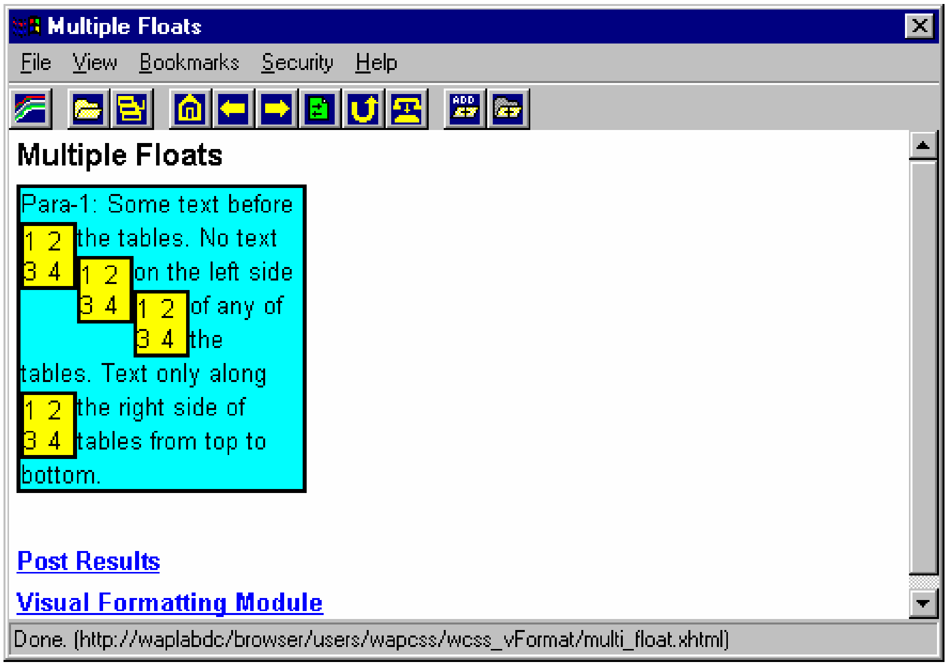
45.25.25 Multiple Floats wcss_vFormat/multi_float.xhtml

Assertion: If the current box is left-floating, and there are any left floating boxes generated by elements earlier in the source document, then for each such earlier box, either the left outer edge of the current box must be to the right of the right outer edge of the earlier box, or its top must be lower than the bottom of the earlier box.

Traceability:

[WAPCSS]	Section 16.2
[CSS2]	Section 9.5.1

Verification: Verify that the "Para-1" has four tables floating to the left. Verify that the text of "Para-1" flows along the right side of tables. Note that depending on the screen width, the four tables may be arranged in one line horizontally or may be arranged in multiple lines and they may extend beyond the bottom borderline of Para-1. But text of "Para-1" must not flow on the left side of any of the tables. For each of left floating boxes preceded by another left floating box, verify that either its left edge is positioned to the right of the right edge of the preceding left floating box or its top edge is positioned lower than the bottom of the preceding left floating box.



Multiple Floats

Para-1: Some text before the tables. No text on the left side of any of the tables. Text only along the right side of tables from top to bottom.

Post Results

Visual Formatting Module

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/multi_float.xhtml)

45.25.26 Clear Left 2 **wcss_vFormat/clear/clear_left2.xhtml**

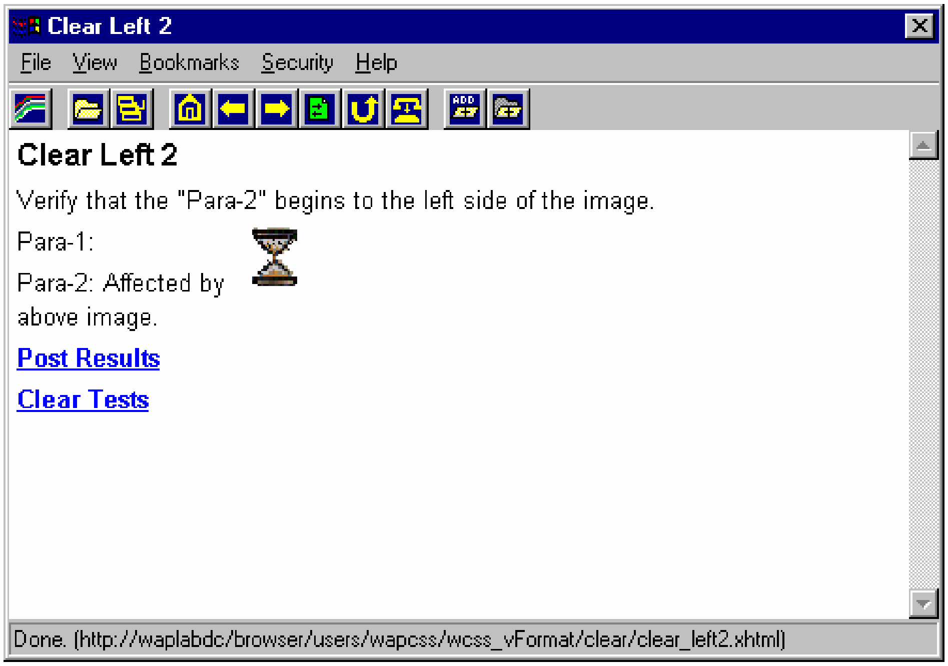
Assertion 1: When the CLEAR property is set to 'left' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any left-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Assertion 2: When the CLEAR property is set to 'right' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any right-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the position of right floating image of "Para-1" affects the display of "Para-2". This means that "Para-2" begins at the left side of image.



Clear Left 2

Verify that the "Para-2" begins to the left side of the image.

Para-1:

Para-2: Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_left2.xhtml)

45.25.27 Clear Right 2 wcss_vFormat/clear/clear_right2.xhtml

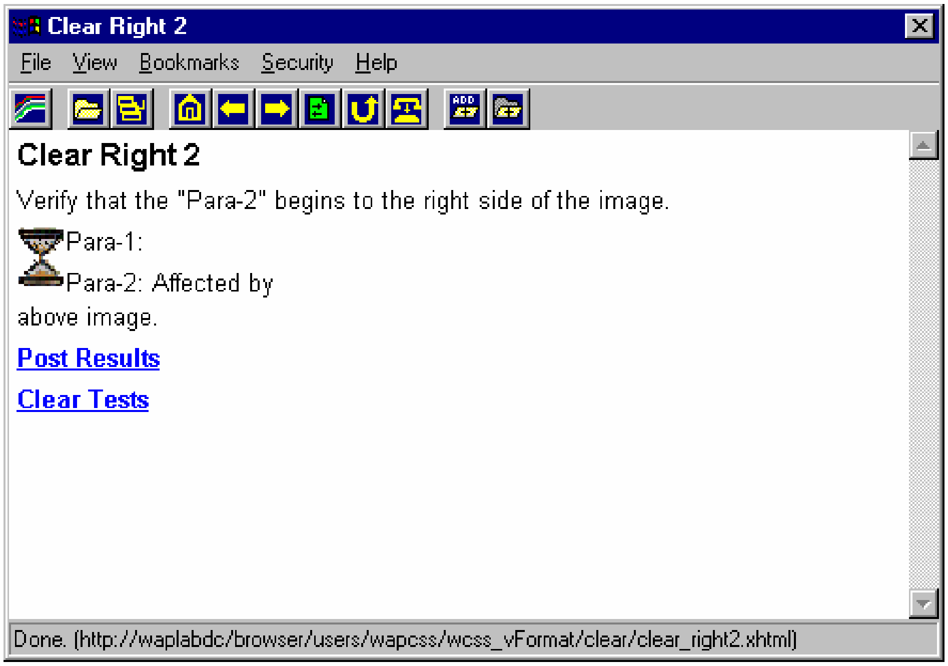
Assertion 1: When the CLEAR property is set to 'left' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any left-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Assertion 2: When the CLEAR property is set to 'right' for an element in a style sheet, the user agent must display the box below the bottom outer edge of any right-floating boxes that resulted from elements earlier in the source document for each instance of the element in the document tree.

Traceability:

[WAPCSS]	Section 16.3
[CSS2]	Section 9.5.2

Verification: Verify that the position of left floating image of "Para-1" affects the display of "Para-2". This means that "Para-2" begins at the right side of image.



Clear Right 2

Verify that the "Para-2" begins to the right side of the image.

Para-1:

Para-2: Affected by above image.

Post Results

Clear Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_vFormat/clear/clear_right2.xhtml)

16 WAP CSS EXTENSION: MARQUEE

This module contains test cases for **"-wap-marquee**" and related properties. Start pages are described below (45.26.01 to 45.26.16). Each test in this section refers to SCR: WCSS-WAPEXT-MARQUEE-C-001.

16.1 Display: –wap-marquee

45.26.01 **Simple Marquee** **wcss_wap_marquee/simple_marquee.xhtml**

Assertion 1: When the DISPLAY property for an element is set to '-wap-marquee' in an inline style attribute, the user-agent must generate one marquee box.

Assertion 2: When the DISPLAY property for an element is set to '-wap-marquee', the user-agent must generate one marquee box with all marquee style properties set to their respective initial values.

Traceability of above 2 assertions:

[WAPCSS]	Section 17.1

Assertion 3: When the -wap-marquee-style property of marquee display is not set, it must default to 'scroll' and hence the text of the marquee box must start completely off one side, scroll all the way across and completely off, and then start again.

Traceability:

[WAPCSS]	Section 17.2

Assertion 4: When no value has been specified for the -wap-marquee-loop property, the marquee must stop scrolling after 1 loop and must be rendered normally.

Traceability:

[WAPCSS]	Section 17.3

Assertion 5: When -wap-marquee-dir property value is not set, it defaults to right to left, independent of the writing direction of the blocks.

Traceability:

[WAPCSS]	Section 17.4

Assertion 6: When the -wap-marquee-speed property value is not specified, it must default to normal speed.

Traceability:

[WAPCSS]	Section 17.5

Verification: Verify that the text "Open Mobile Alliance" scrolls across the screen as a marquee. Verify that it starts completely off on right side, scrolls all the way across to left and completely off. Verify that it scrolls only one loop at normal speed.

45.26.02 Picto Marquee wcss_wap_marquee/picto_marquee.xhtml

Assertion: When the content of the marquee box is pictograms and when '-wap-marquee-style' property is set, the device must display the marquee content with the set marquee style.

Traceability:

[WAPCSS]	Section 17.2

Verification: Verify that the text "Open Mobile Alliance" scrolls across the screen as a marquee and verify that marquee also contains an image.

16.2 Marquee Style: '-wap-marquee-style'

45.26.03 Style Scroll wcss_wap_marquee/style_scroll.xhtml

Assertion 1: When the -wap-marquee-style property of marquee display is set to 'scroll', the text of the marquee box must start completely off one side, scroll all the way across and completely off, and then start again.

Assertion 2: When the content of the marquee box is plain text and when '-wap-marquee-style' property is set, the device must display the marquee content with the set marquee style.

Assertion 3: When the content of the marquee box is plain text and when '-wap-marquee-style' property is set, the device must display the marquee content with the set marquee style.

Traceability of above 3 assertions:

[WAPCSS]	Section 17.2

Verification: Verify that the marquee "Open Mobile Alliance" starts completely off on right side, scrolls all the way across to left and completely off. Verify that the marquee loops four times altogether. Verify that the marquee text is not displayed on the screen, at the end of all loops.

45.26.04 Style Slide wcss_wap_marquee/style_slide.xhtml

Assertion: When the -wap-marquee-style property of marquee display is set to 'slide', the text of the marquee box must start completely off one side, scroll in and stop as soon as the text touches the other margin.

Traceability:

[WAPCSS]	Section 17.2

Verification: Verify that the marquee "Open Mobile Alliance" starts completely off on right side, scrolls all the way across to left and stops as soon as the text touches the left margin. Verify that the marquee loops four times altogether. Verify that the marquee text remains displayed on the screen, at the end of all loops.

45.26.05 **Style Alternate** **wcss_wap_marquee/style_alt.xhtml**

Assertion: When the -wap-marquee-style property of marquee display is set to 'alternate', the text of the marquee box must start on one side, scroll in, stop as soon as the text touches the other margin and then start scrolling in the opposite direction. Again as soon as the text touches other margin it should change the direction.

Traceability:

[WAPCSS]	Section 17.2

Verification: Verify that the marquee "Open Mobile Alliance" starts on right side, scrolls in and stops as soon as it touches the other end of the screen and then starts scrolling in opposite direction. Verify that every time it touches the other end of the screen, it changes the direction. Verify that the marquee loops four times altogether. Verify that the marquee text remains displayed on the screen, at the end of all loops.

16.3 Marquee Loop: '-wap-marquee-loop'

45.26.06 Marquee Loops **wcss_wap_marquee/loop_3.xhtml**

Assertion 1: When the loop value is set to a valid value, the text must scroll across the screen that many times.

Assertion 2: When the loop value has reached its maximum limit, the text must stop scrolling and must be rendered normally.

Traceability of above 2 assertions:

[WAPCSS]	Section 17.3

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

Traceability:

[WAPCSS]	Section 6.2
[CSS2]	Section 4.3.1

Verification: Verify that the marquee text "Open Mobile Alliance" loops 3 times.

45.26.07 Max Loops **wcss_wap_marquee/loop_max.xhtml**

Assertion 1: The device must support at least 16 iterations for the -wap-marquee-loop.

Assertion 2: When the loop value has reached its maximum limit, the text must stop scrolling and must be rendered normally.

Traceability of above 2 assertions:

[WAPCSS]	Section 17.3

Verification: Verify that the marquee text "Open Mobile Alliance" loops at least 16 times.

45.26.08 Zero Loops **wcss_wap_marquee/loop_zero.xhtml**

Assertion: When the loop value is set to 0, the text must not scroll and still be visible to the user.

Traceability:

[WAPCSS]	Section 17.3

Verification: Verify that the marquee text "Open Mobile Alliance" does not scroll at all and still be visible.

16.4 Marquee Direction: '-wap-marquee-dir'

45.26.09 Left to Right **wcss_wap_marquee/dir_ltr.xhtml**

Assertion: When the -wap-marquee-dir property value is set to 'ltr', the text must scroll from left to right.

Traceability:

[WAPCSS]	Section 17.4

Verification: Verify that the marquee text "Open Mobile Alliance" scrolls from left to right eight times.

45.26.10 Right to Left **wcss_wap_marquee/dir_rtl.xhtml**

Assertion: When -wap-marquee-dir value is set to 'rtl', the text must scroll from right to left.

Traceability:

[WAPCSS]	Section 17.4

Verification: Verify that the marquee text "Open Mobile Alliance" scrolls from right to left eight times.

16.5 Marquee Speed: '-wap-marquee-speed'

45.26.11 Marquee Speeds wcss_wap_marquee/speeds.xhtml

Assertion: When the -wap-marquee-speed property value is set to 'slow', 'normal' and 'fast', the relation slow < normal < fast must be true.

Traceability:

[WAPCSS]	Section 17.5

Verification: Verify that three marquees are displayed which scrolls at different speeds. Verify that marquee "slow marquee" scrolls slower than "normal marquee". Verify that marquee "fast marquee" scrolls faster than "normal marquee".

45.26.12 Invalid Speeds wcss_wap_marquee/inv_speeds.xhtml

Assertion: When the -wap-marquee-speed property value is set to an invalid value, then the initial value normal must be used for scrolling speed of the marquee.

Traceability:

[WAPCSS]	Section 17.5

Verification: Verify that both the marquees "Normal Speed 1" and "Normal Speed 2" scrolls at normal speed only.

16.6 More Marquee Tests

45.26.13 Long Slide **wcss_wap_marquee/long_slide.xhtml**

Assertion: When the -wap-marquee-style property of marquee display is set to 'slide' and when the content length of marquee is longer than its block width, then in the last loop, the marquee text must scroll in and stop as soon as the end of the text is visible on the screen.

Traceability:

[WAPCSS]	Section 17.2.3

Verification: Verify that the marquee "Wish you very happy and prosperous new year!" starts completely off on right side of its containing box, scrolls all the way across to left and stops as soon as the last character of the marquee is visible in the containing box at right edge.

45.26.14 Long Alternate 1 **wcss_wap_marquee/long_alt1.xhtml**

Assertion: When the -wap-marquee-style property of marquee display is set to 'alternate' and when the content length of marquee is longer than its block width, then the marquee text must start completely off on one side, scroll in till the last character appears on the screen, stop and reverse its direction till the first character appears on the screen, stop and reverse its direction till it completes the number of set loops.

Traceability:

[WAPCSS]	Section 17.2.3

Verification: Verify that the marquee "Wish you very happy and prosperous new year!" starts completely off on right side, scrolls all the way across to left and stops as soon as the last character of the marquee is visible on the screen at right edge of its containing box. Verify that then it reverses its direction and scrolls towards right till the first character of marquee appears on left side. Verify that after completing 2 loops, the beginning part of the marquee text remains displayed in its containing block.

45.26.15 Long Alternate 2 **wcss_wap_marquee/long_alt2.xhtml**

Assertion: When the -wap-marquee-style property of marquee display is set to 'alternate' and when the content length of marquee is longer than its block width, then the marquee text must start completely off on one side, scroll in till the last character appears on the screen, stop and reverse its direction till the first character appears on the screen, stop and reverse its direction till it completes the number of set loops.

Traceability:

[WAPCSS]	Section 17.2.3

Verification: Verify that at the end of 3 loops, the end part of marquee text "Wish you very happy and prosperous new year!" remains displayed in its containing block.

45.26.16 Invalid Loops **wcss_wap_marquee/inv_loops.xhtml**

Assertion 1: When the -wap-marquee-loop value is assigned to an invalid value, then the initial value 1 must be used and marquee must stop after scrolling one loop.

Traceability:

[WAPCSS]	Section 17.3

Assertion 2: When the -wap-marquee-style value is assigned to an invalid value, then the initial value 'scroll' must be used and the text of the marquee box must start completely off one side, scroll all the way across and completely off, and then start again.

Traceability:

[WAPCSS]	Section 17.2

Verification: Verify that at the marquee text "Marquee 1" scrolls across screen only once. Verify that the marquee text "Marquee 2" loops three times starting completely off on right side scrolling all the way across to left and getting completely off.

17 WAP CSS EXTENSION: ACCESS KEYS

This module contains test cases for "**-wap-accesskey**" property. Start pages are described below (45.27.00 to 45.27.08). Each test in this section refers to SCR: WCSS-WAPEXT-ACC-C-001.

17.1 Access Keys: '-wap-accesskey'

45.27.00 Invalid key **wcss_wap_accesskey/akey_invalid.xhtml**

Assertion 1: When no value is specified for the –wap-accesskey property, it must not be set to any access key.

Traceability:

[WAPCSS]	Section 18.1

NOTE: This assertion is untestable, as it is not feasible to test for all the keys or combination of keys available on a device.

Assertion 2: When an invalid value is assigned to the –wap-accesskey property, the user agent must ignore it.

Traceability:

[WAPCSS]	Section 18.1

Verification: Verify that none of the keys "1" or "2" activate the link "Page 1".

17.1.1 Access Key Availability and Assignment

45.27.01 **Availability** **wcss_wap_accesskey/akey_available.xhtml**

Assertion 1: When an accesskey is available for an element, the user agent must use the specified access key and assign it to the element.

Assertion 2: When an accesskey has already been assigned to an element, it must become unavailable for another element.

Assertion 3: The user agent should render the value of the access key in such a way as to emphasize its role and to distinguish it from other characters.

Traceability of above 3 assertions:

[WAPCSS]	Section 18.1.1

Verification: Verify that key "1" activates the "Page 1" link and key "2" activates the "Page 2" link. Verify that key "1" does not activate the "Page 3" link.

(When any of the "Page 1", "Page 2" or "Page 3" is loaded, activate the "Back" link on that page to get back to the test case start page.)

Verify that the values of access keys are rendered in such a way to distinguish them from other characters.

17.1.2 Fallbacks and Multiple Assignments

45.27.02 **Comma Separated** **wcss_wap_accesskey/akey_comma.xhtml**

Assertion: When several ',' separated values are assigned to an accesskey, the user agent must select the first available access key in the list for the assignment and ignore all others.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: The comma separated key list "1,2,3" is assigned to "Page 1" link and list "1,4,5" is assigned to "Page 2" link in this page.

Verify that out of the assigned keys 1,2,3 only key "1" activates the "Page 1" link and out of the assigned keys 1,4,5 only key "4" activates the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.03 **Space Separated** **wcss_wap_accesskey/akey_space.xhtml**

Assertion: When several spaces separated values are assigned to an accesskey, the user agent must select all available access keys from the list for the assignment.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: The space separated key list "1 2 3" is assigned to "Page 1" link and list "2 3 4 5" is assigned to "Page 2" link in this page.

Verify that all of the assigned access keys 1,2,3 activate the "Page 1" link. Verify that out of the assigned keys 2, 3, 4, 5 only keys "4" and "5" activate the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.04 **Space and Comma Separated** **wcss_wap_accesskey/akey_comspace.xhtml**

Assertion: When the space and comma separated values are used together to assign values to an access key, then the user agent must first try to assign all of the available values from space separated values and if none of them are available, must assign the first available value from the comma separated values.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: The space and comma separated key list "1 2, 3, 4" is assigned to "Page 1" link and the list "2 3, 4, 5" is assigned to "Page 2" link in this page.

Verify that out of the assigned keys 1,2,3,4, only keys "1" and "2" activate the "Page 1" link. Verify that out of the assigned keys 2, 3, 4, 5 only key "3" activates the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.05 **accesskey Attribute** **wcss_wap_accesskey/akey_attribute.xhtml**

Assertion: When the XHTML accesskey attribute is present on an element it must be translated into the -wap-accesskey rule with specificity equal to zero.

Traceability:

[WAPCSS]	Section 18.1.3
[XHTMLBasic]	Section 3

Verification: Verify that out of keys 1,2 only the access key "2" activates the "Page 1" link.

(When the "Page 1" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.06 **Space and Comma 2** **wcss_wap_accesskey/akey_comspace2.xhtml**

Assertion: When the space and comma separated values are used together to assign values to an access key, then the user agent must first try to assign all of the available values from space separated values and if none of them are available, must assign the first available value from the comma separated values.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: The space and comma separated key list "1 2, 3, 4" is assigned to "Page 1" link and the list "1 2, 4, 5" is assigned to "Page 2" link in this page.

Verify that out of the assigned keys 1,2,3,4, only keys "1" and "2" activate the "Page 1" link. Verify that out of the assigned keys 1, 2, 4, 5 only key "4" activates the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.07 **Space and Comma 3** **wcss_wap_accesskey/akey_comspace3.xhtml**

Assertion: When the space and comma separated values are used together to assign values to an access key, then the user agent must first try to assign all of the available values from space separated values and if none of them are available, must assign the first available value from the comma separated values.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: The space and comma separated key list "1 2, 3, 4" is assigned to "Page 1" link and the list "1, 2, 4, 5" is assigned to "Page 2" link in this page.

Verify that out of the assigned keys 1,2,3,4, only keys "1" and "2" activate the "Page 1" link. Verify that out of the assigned keys 1, 2, 4, 5 only key "4" activates the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

45.27.08 **Other keys** **wcss_wap_accesskey/akey_others.xhtml**

Assertion: When the space and comma separated values are used together to assign values to an access key, then the user agent must first try to assign all of the available values from space separated values and if none of them are available, must assign the first available value from the comma separated values.

Traceability:

[WAPCSS]	Section 18.1.2

Verification: Verify that only send and star (*) keys activate the "Page 1" link. Verify that only hash (#) key "4" activates the "Page 2" link.

(When either of the "Page 1" or "Page 2" is loaded, activate the "Back" link on that page to get back to the test case start page.)

18 WAP CSS EXTENSION: INPUT

18.1 General Concepts

This module contains the test cases numbered 45.28.01 to 45.28.22 for testing the –wap-input-format and –wap-input-required properties. Each test in this section refers to SCR: WCSS-WAPEXT-INPUT-C-001.

18.2 Input-Format: '-wap-input-format'

This sub-module contains test cases for "-wap-input-format" property. Start pages are described below (45.28.01 to 45.28.13).

45.28.01 Format "*A" wcss_wap_input/input_format/wmlformat_upper_A.xhtml

Assertion: When the -wap-input-format property for an input element is specified as "*A", the text input must be limited to upper-case letter, symbol or punctuation characters.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to uppercase letters, symbols, or punctuation characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.02 Format "*a" wcss_wap_input/input_format/wmlformat_lower_a.xhtml

Assertion: When the -wap-input-format property of the input element is specified as "*a", the text input is limited to lower-case letter, symbol or punctuation characters.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to lowercase letters, symbols or punctuation characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.03 **Format "*N"** **wcss_wap_input/input_format/wmlformat_upper_N.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*N", the text input is limited to numeric characters.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to numeric characters only. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.04 **Format "*n"** **wcss_wap_input/input_format/wmlformat_lower_n.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*n", the text input is limited to any numeric, symbol, punctuation character.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to any numeric, symbol or punctuation characters only. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.05 **Format "*X"** **wcss_wap_input/input_format/wmlformat_upper_X.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*X", the text input is limited to any upper-case letter, numeric character, symbol or punctuation character.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to any uppercase letters, numeric characters, symbols or punctuation characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.06 **Format "*x"** **wcss_wap_input/input_format/wmlformat_lower_x.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*x", the text input is limited to any lower-case letter, numeric character, symbol or punctuation character.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the input text box is limited to any lowercase letters, numeric characters, symbols or punctuation characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.07 **Format "*M"** **wcss_wap_input/input_format/wmlformat_upper_M.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*M", the text input can be any character, but the user-agent may assume that the input is in upper case.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box can be any character. The UA may assume that the input is in upper case. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.08 **Format "*m"** **wcss_wap_input/input_format/wmlformat_lower_m.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "*m", the text input can be any character, but the user-agent may assume that the input is in lower-case.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box can be any character. The UA may assume that the input is in lower-case. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.09 **Invalid Format** **wcss_wap_input/input_format/wmlformat_invalid.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as an invalid mask, the mask must be ignored by the user agent and must default to '*M' (must accept any characters as input, but the user-agent may assume that the input is in upper-case.).

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box can be any character. The UA may assume that the input is in upper case. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.10 **Default Format** **wcss_wap_input/input_format/wmlformat_none.xhtml**

Assertion: When the -wap-input-format property of the input element is set with no value it must default to '*M' and the text input can be any character, but the user-agent may assume that the input is in upper case.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box can be any character. The UA may assume that the input is in upper case. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.11 **Format "nf"** **wcss_wap_input/input_format/wmlformat_lower_5a.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as "nf", where n is a natural number and f is one of the format characters "aANXx", the number of input characters is limited to n and the type is limited to f.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input text box is limited to up to 5 lowercase letters, symbols or punctuation characters. Verify that "Para-1" is displayed normally. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.12 Multiple Format wcss_wap_input/input_format/wmlformat_NNAANNaa.xhtml

Assertion: When the -wap-input-format property of the input element is specified as a series of format characters, the input of each position in the input field is limited as specified.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the user agent limits the input to the mask "NNAANNaa", i.e. 2 numeric characters; followed by 2 uppercase chars, symbols or punctuation marks; then 2 numeric characters; then two lowercase chars, symbols or punctuation marks. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.13 Escaped Chars wcss_wap_input/input_format/wmlformat_escaped_characters.xhtml

Assertion: When the -wap-input-format property of the input element is specified as two backslashes "\\" followed by a character c, that character is displayed in the input field at the position indicated.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Verification: Verify that the entry into the input field is limited to 10 numeric characters, in the format NNN-NNN-NNNN. Verify that the '-' are pre-displayed in the input field at 4th and 8th positions. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form along with the '-' characters. Post results from this page.

18.2.1 The WML 'format' attribute

Assertion: When the WML 'format' attribute is present on an element it must be translated into the –wap-input-format rule with specificity equal to zero.

Traceability:

[WAPCSS]	Section 19.2.1

NOTE: This assertion is not applicable to present test suite, as this is applicable only to a WML document, but not to XHTMLBasic documents.

18.3 Required Input: '-wap-input-required'

This sub-module contains test cases for "-wap-input-required" property. Start pages are described below (45.28.14 to 45.28.16)

45.28.14 Input Required True
wcss_wap_input/input_rqd/input_required_true.xhtml

Assertion: When the ' -wap-input-required' property of the input element is specified as "true", an input must be provided to the input element.

Traceability:

[WAPCSS]	Section 19.3

Verification: Verify that the input field requires an input by leaving the field blank and attempting to submit. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.15 Input Required False
wcss_wap_input/input_rqd/input_required_false.xhtml

Assertion: When '-wap-input-required' is set to "false", the input can be left blank.

Traceability:

[WAPCSS]	Section 19.3

Verification: Verify that the input field does NOT require an input by leaving the field blank and attempting to submit. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.16 No Input Required
wcss_wap_input/input_rqd/no_input_required.xhtml

Assertion: When the ' -wap-input-required' property of the input element is not set, the input can be left blank.

Traceability:

[WAPCSS]	Section 19.3

Verification: Verify that the input field does NOT require an input by leaving the field blank and attempting to submit. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

18.3.1 Input Format and Input Required Combination

This sub-module contains test cases for "-wap-input-format" and "-wap-input-required" properties used together. Start pages are described below. (45.28.17 to 45.28.20)

45.28.17 Required - Format: NNN

wcss_wap_input/input_both/required_format_nonzero.xhtml

Assertion: When the ' -wap-input-required' property of the input element is specified as "true" and '-wap-input-format' requires a non-zero length string input (eg: NNN), then zero length input is not allowed and the input must be provided as per the '-wap-input-format'.

Traceability:

[WAPCSS]	Section 19.3.1

Verification: Verify that the input field accepts exactly 3 numeric characters only, and will not accept blank input. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.18 Required - Format: *N

wcss_wap_input/input_both/required_format_zero.xhtml

Assertion: When the ' -wap-input-required' property of the input element is specified as "true" and '-wap-input-format' allows a zero length string input (eg: *M), then zero length input is not allowed and the input must be provided as per the '-wap-input-format'.

Traceability:

[WAPCSS]	Section 19.3.1

Verification: Verify that the input field accepts any number of numeric characters only, and will not accept blank input. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.19 Not Required - Format: NNN

wcss_wap_input/input_both/not_required_format_nonzero.xhtml

Assertion: When the ' -wap-input-required' property of the input element is specified as "false" and '-wap-input-format' requires a non-zero length string input (eg: NNN), then zero length input is allowed and any other input must be provided as per the '-wap-input-format'.

Traceability:

[WAPCSS]	Section 19.3.1

Verification: Verify that the input field accepts either blank input or exactly three numeric characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.20 Not Required - Format: *N wcss_wap_input/input_both/not_required_format_zero.xhtml

Assertion: When the ' -wap-input-required' property of the input element is specified as "false" and '-wap-input-format' allows a zero length string input (eg: *M), then zero length input is allowed and any other input must be provided as per the '-wap-input-format'.

Traceability:

[WAPCSS]	Section 19.3.1

Verification: Verify that the input field accepts either blank input or any number of only numeric characters. Submit the form to load page tilted "Form Response" which displays the input entered in previous page form. Post results from this page.

45.28.21 Escape Chars 2 wcss_wap_input/input_format/esc_chars2.xhtml

Assertion 1: When the -wap-input-format property of the input element is specified as two backslashes "\\" followed by a character c, that character is displayed in the input field at the position indicated.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

Assertion 2: 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 entry into the input field is limited to 2 numeric characters, in the format NδN. Verify that the Greek character Delta is pre-displayed in the input field at 2nd position.

45.28.22 **Escape Chars 3** **wcss_wap_input/input_format/esc_chars3.xhtml**

Assertion: When the -wap-input-format property of the input element is specified as two backslashes "\\" followed by a character c, that character is displayed in the input field at the position indicated.

Traceability:

[WAPCSS]	Section 19.2
[WML13]	Section 6.5.2.1

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 entry into the input field is limited to 2 numeric characters, in the format NθN. Verify that the Greek character Theta is pre-displayed in the input field at 2nd position.

18.3.2 The WML 'emptyok' Attribute

Assertion: When the WML 'emptyok' attribute is present on an element it must be translated into the –wap-input-required rule with specificity equal to zero.

Traceability:

[WAPCSS]	Section 19.2.1

NOTE: This assertion is not applicable to present test suite, as this is applicable only to a WML document, but not to XHTMLBasic documents.

19 INHERITANCE TESTS

When cascade results in a value for a property of an element, then that property of element is assigned the cascaded value. Otherwise the inheritance comes into picture.

Some properties (eg: color) inherit the value from its parent element by default. Some properties (eg: border-width) don't inherit values from their parent element by default. These properties are assigned the initial value defined for them. However, if the property value is explicitly set to 'inherit', then it forces the element to inherit the computed value from its parent element for that property.

This section covers each and every property of [WAPCSS] for default inheritance, initial values and for forcing inheritance by setting property value as 'inherit'. Every test in this section (45.29.01 to 45.29.63) refers to SCR: WCSS-CASC-C-001.

19.1 Margin

45.29.01 **Margin-Top** **wcss_inherit/margin/mtop_inherit1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

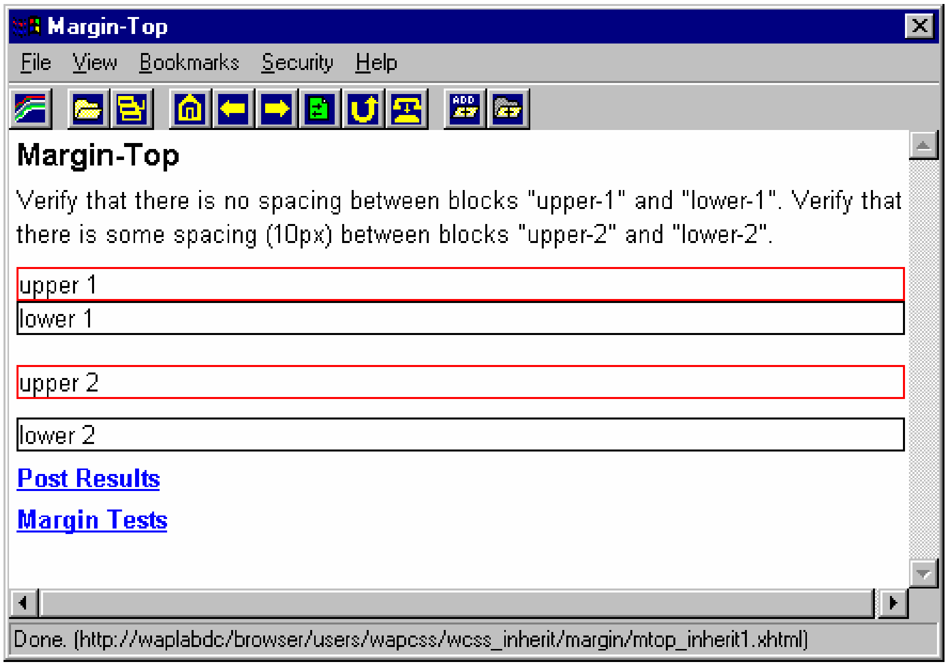
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is no spacing between blocks "upper-1" and "lower-1". Verify that there is some spacing (10px) between blocks "upper-2" and "lower-2".



Margin-Top

Verify that there is no spacing between blocks "upper-1" and "lower-1". Verify that there is some spacing (10px) between blocks "upper-2" and "lower-2".

upper 1

lower 1

upper 2

lower 2

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/margin/mtop_inherit1.xhtml)

45.29.02 **Margin-Bottom** **wcss_inherit/margin/mbott_inherit1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

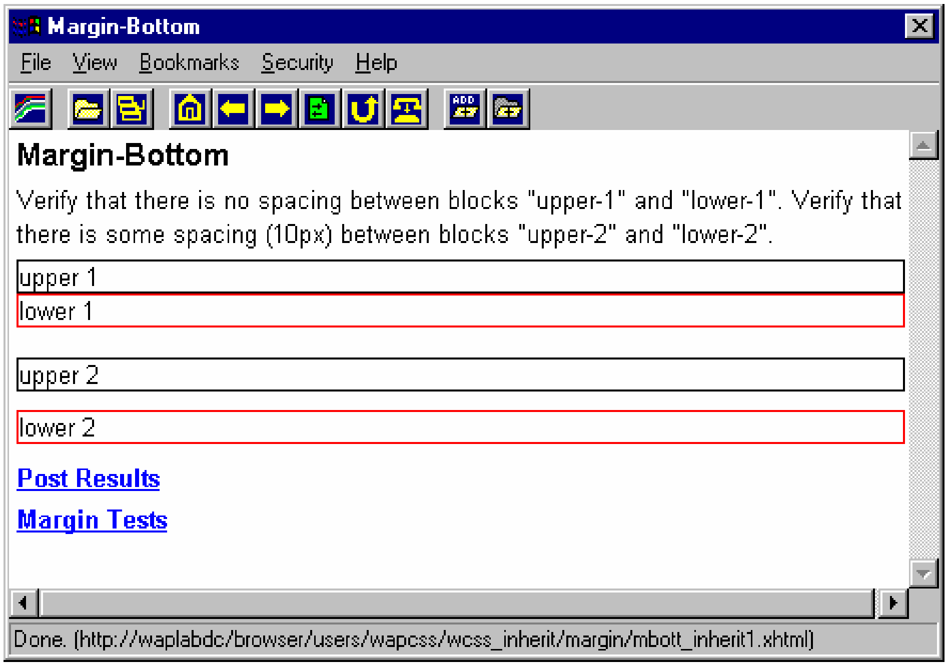
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is no spacing between blocks "upper-1" and "lower-1". Verify that there is some spacing (10px) between blocks "upper-2" and "lower-2".



Margin-Bottom

Verify that there is no spacing between blocks "upper-1" and "lower-1". Verify that there is some spacing (10px) between blocks "upper-2" and "lower-2".

upper 1

lower 1

upper 2

lower 2

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/margin/mbott_inherit1.xhtml)

45.29.03 **Margin-Left** **wcss_inherit/margin/mleft_inherit1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

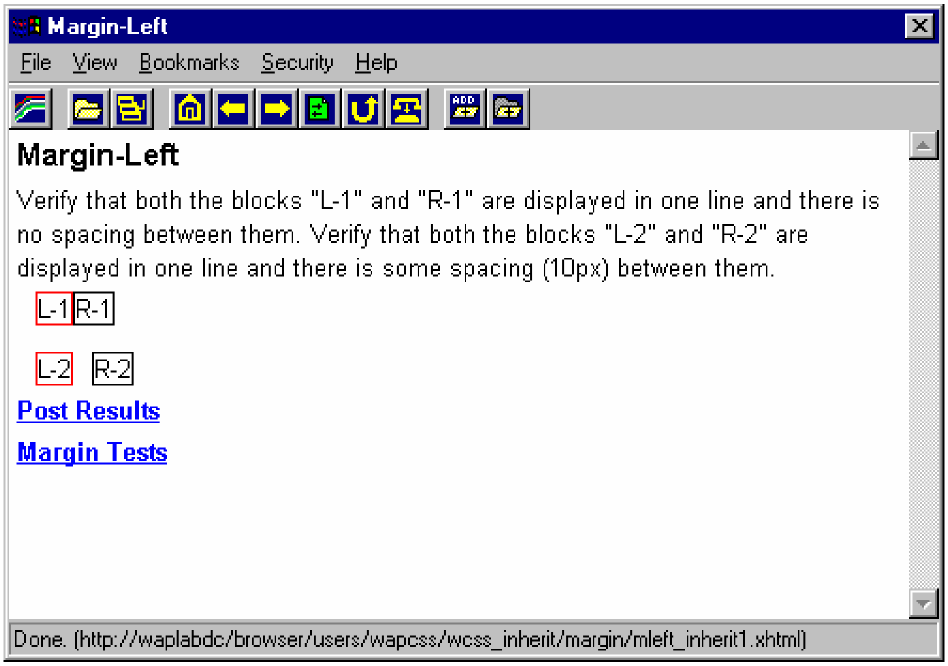
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that both the blocks “L-1” and “R-1” are displayed in one line and there is no spacing between them. Verify that both the blocks "L-2" and "R-2" are displayed in one line and there is some spacing (10px) between them.



Margin-Left

Verify that both the blocks "L-1" and "R-1" are displayed in one line and there is no spacing between them. Verify that both the blocks "L-2" and "R-2" are displayed in one line and there is some spacing (10px) between them.

L-1R-1

L-2 R-2

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/margin/mleft_inherit1.xhtml)

45.29.04 **Margin-Right** **wcss_inherit/margin/mright_inherit1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

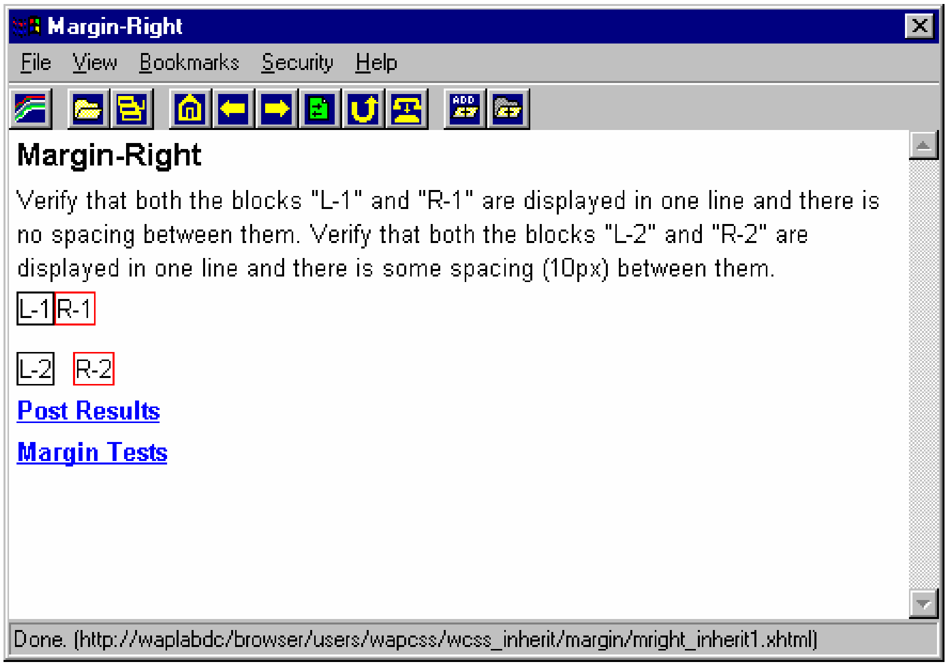
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is no horizontal spacing (gap) between right edge of block "Left 1" and left edge of block "Right 1". Verify that there is some horizontal spacing (gap) between right edge of block "Left 2" and left edge of block "Right 2".



Margin-Right

Verify that both the blocks "L-1" and "R-1" are displayed in one line and there is no spacing between them. Verify that both the blocks "L-2" and "R-2" are displayed in one line and there is some spacing (10px) between them.

L-1 R-1

L-2 R-2

Post Results

Margin Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/margin/mright_inherit1.xhtml)

45.29.05 Margin Inherit **wcss_inherit/margin/margin_inherit.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

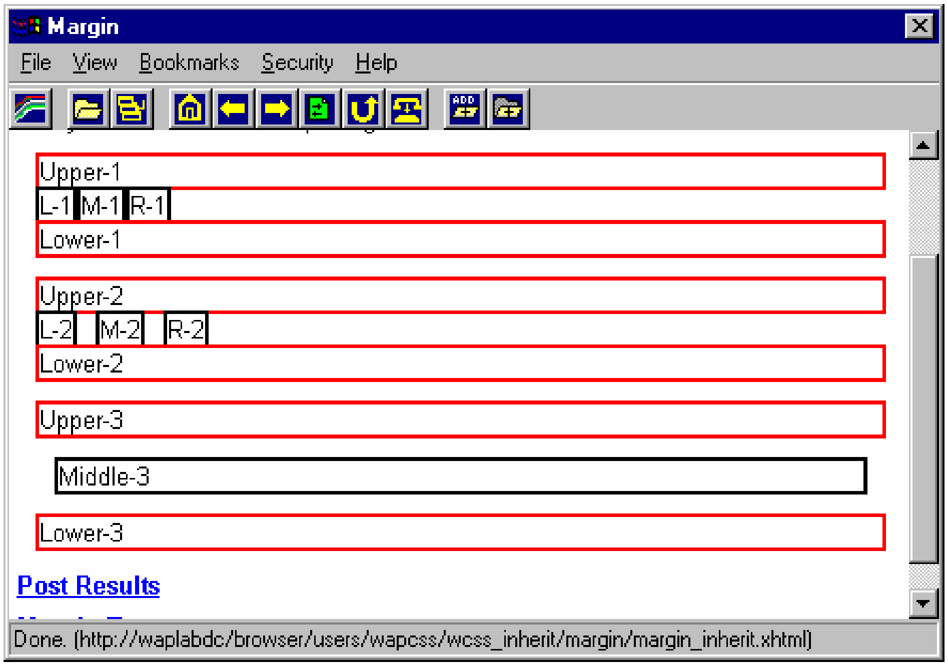
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is no spacing between "M-1" and its surrounding blocks. Verify that there is some spacing on both left and right sides of block "M-2". Verify that there is some spacing on all four sides of the block "Middle-3".



Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/margin/margin_inherit.xhtml)

19.2 Padding

45.29.06 Padding-Top wcss_inherit/padding/padtop_inherit2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

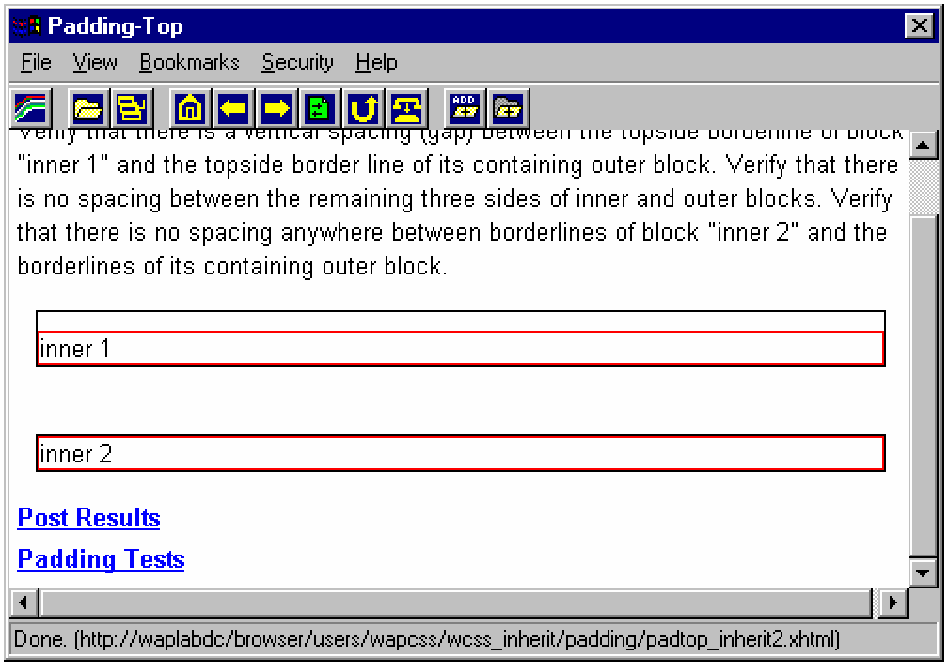
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical spacing (gap) between the topside borderline of block "inner 1" and the topside border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.



Verify that there is a vertical spacing (gap) between the topside borderline of block "inner 1" and the topside border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.

inner 1

inner 2

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/padding/padtop_inherit2.xhtml)

45.29.07 **Padding-Bottom** **wcss_inherit/padding/padbot_inherit2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

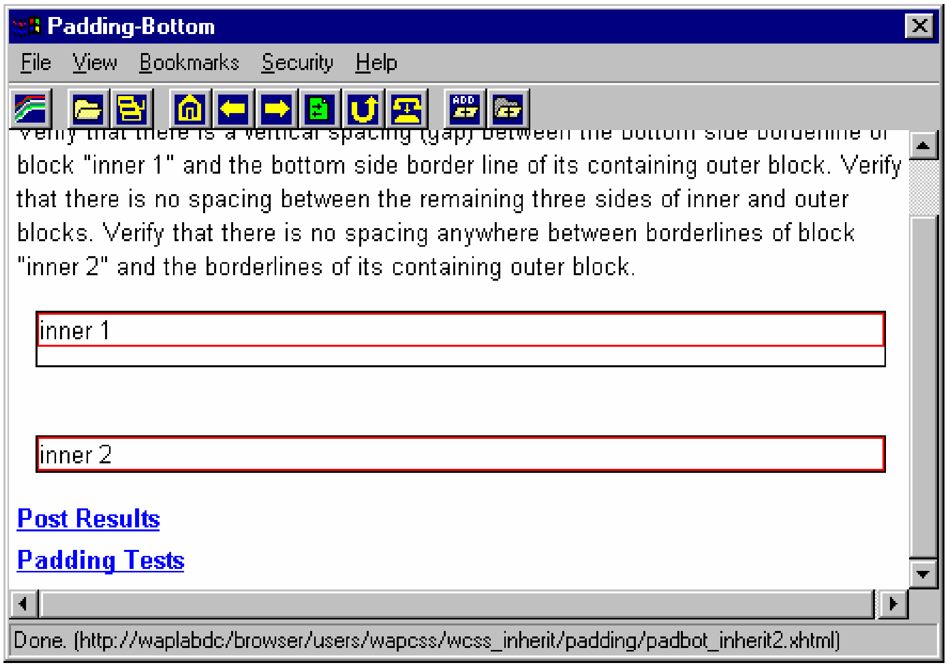
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical spacing (gap) between the bottom side borderline of block "inner 1" and the bottom side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.



Verify that there is a vertical spacing (gap) between the bottom side borderline of block "inner 1" and the bottom side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.

inner 1

inner 2

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/padding/padbot_inherit2.xhtml)

45.29.08 **Padding-Left** **wcss_inherit/padding/padleft_inherit2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

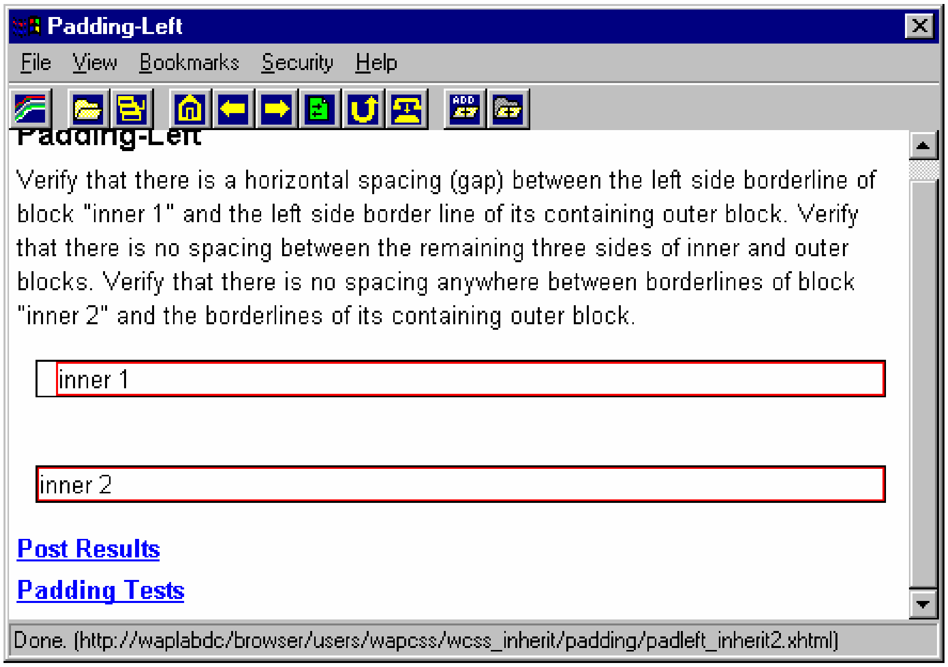
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a horizontal spacing (gap) between the left side borderline of block "inner 1" and the left side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.



Padding-Left

Verify that there is a horizontal spacing (gap) between the left side borderline of block "inner 1" and the left side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.

inner 1

inner 2

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/padding/padleft_inherit2.xhtml)

45.29.09 **Padding-Right** **wcss_inherit/padding/padright_inherit2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

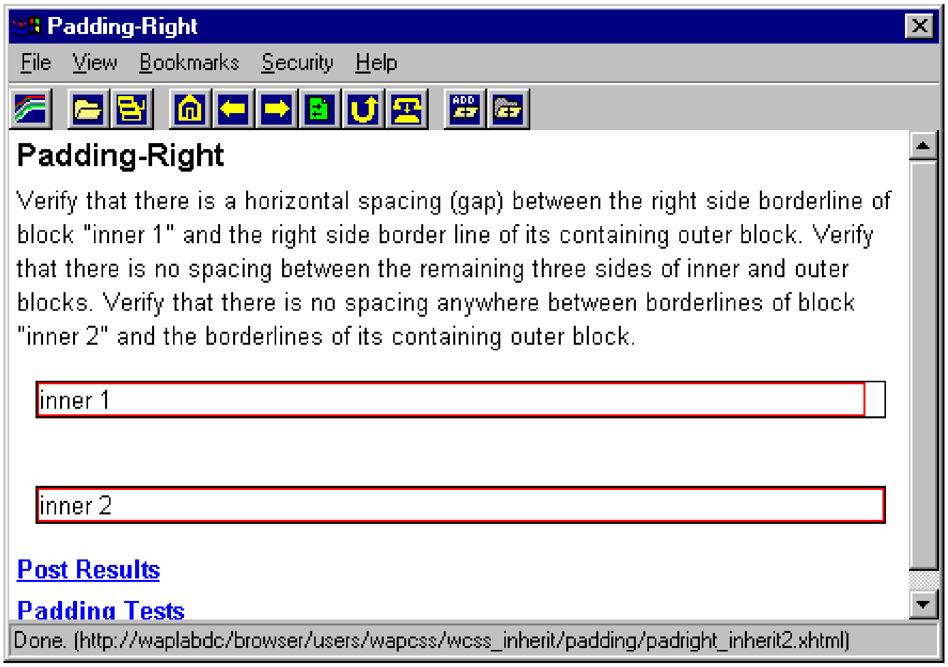
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a horizontal spacing (gap) between the right side borderline of block "inner 1" and the right side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.



Padding-Right

Verify that there is a horizontal spacing (gap) between the right side borderline of block "inner 1" and the right side border line of its containing outer block. Verify that there is no spacing between the remaining three sides of inner and outer blocks. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.

inner 1

inner 2

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/padding/padright_inherit2.xhtml)

45.29.10 Padding **wcss_inherit/padding/padding_inherit.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

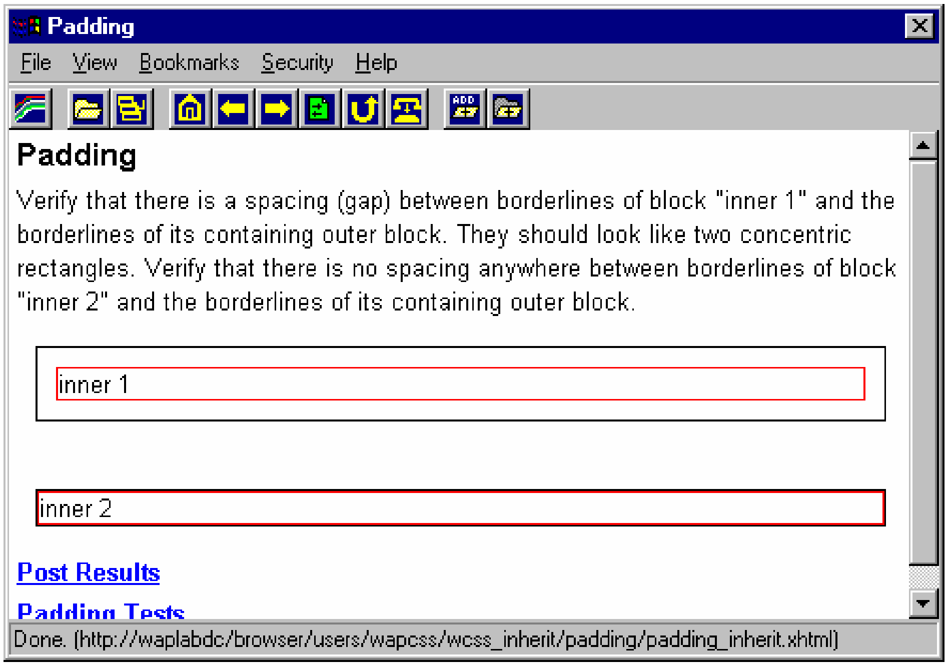
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a spacing (gap) between borderlines of block "inner 1" and the borderlines of its containing outer block. They should look like two concentric rectangles. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.



Padding

Verify that there is a spacing (gap) between borderlines of block "inner 1" and the borderlines of its containing outer block. They should look like two concentric rectangles. Verify that there is no spacing anywhere between borderlines of block "inner 2" and the borderlines of its containing outer block.

inner 1

inner 2

Post Results

Padding Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/padding/padding_inherit.xhtml)

19.3 Border Width

45.29.11 BorderTop Width 1 wcss_inherit/border/bw_top1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick upper borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.12 BorderTop Width 2 wcss_inherit/border/bw_top2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick upper borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.13 **BorderBottom Width 1** **wcss_inherit/border/bw_bott1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick bottom borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.14 **BorderBottom Width 2** **wcss_inherit/border/bw_Bott2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick bottom borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.15 **BorderLeft Width 1** **wcss_inherit/border/bw_left1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick left borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.16 **BorderLeft Width 2** **wcss_inherit/border/bw_left2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick left borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.17 **BorderRight Width 1** **wcss_inherit/border/bw_right1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick right borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.18 **BorderRight Width 2** **wcss_inherit/border/bw_right2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has a thick right borderline and medium size (normal size) lines for other three borders. Verify that the inner "Block-2" has medium size borderlines on all four sides.

45.29.19 Border Width **wcss_inherit/border/bw_all.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the inner "Block-1" has thick borderlines on all four sides and the inner "Block-2" has medium (normal) size borderlines on all four sides.

19.4 Border Color

45.29.20 BorderTop Color 1 wcss_inherit/border/bc_top1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored topside border line and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.21 BorderTop Color 2 wcss_inherit/border/bc_top2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored topside border line and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.22 BorderBottom Color 1 wcss_inherit/border/bc_bott1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored bottom side borderline and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.23 BorderBottom Color 2 wcss_inherit/border/bc_Bott2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored bottom side borderline and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.24 BorderLeft Color 1 wcss_inherit/border/bc_left1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored left side borderline and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.25 BorderLeft Color 2 wcss_inherit/border/bc_left2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored left side borderline and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.26 BorderRight Color 1 wcss_inherit/border/bc_right1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored right side border line and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.27 BorderRight Color 2 wcss_inherit/border/bc_right2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored right side border line and default colored lines for other three sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

45.29.28 **Border Color** **wcss_inherit/border/bc_all.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the "Block-1" has red colored borderlines on all four sides. Verify that the inner "Block-2" has only default colored borderlines on all four sides.

19.5 Border Style

45.29.29 BorderTop Style 1 wcss_inherit/border/bs_top1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

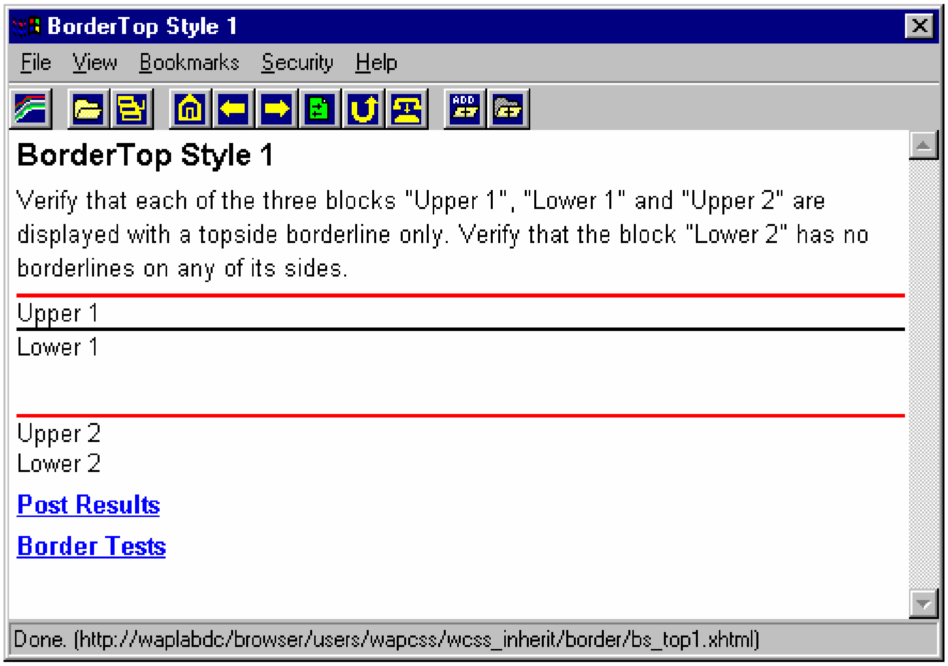
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that each of the three blocks "Upper 1", "Lower 1" and "Upper 2" are displayed with a topside borderline only. Verify that the block "Lower 2" has no borderlines on any of its sides.



BorderTop Style 1

Verify that each of the three blocks "Upper 1", "Lower 1" and "Upper 2" are displayed with a topside borderline only. Verify that the block "Lower 2" has no borderlines on any of its sides.

Upper 1

Lower 1

Upper 2

Lower 2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_top1.xhtml)

45.29.30 BorderTop Style 2 wcss_inherit/border/bs_top2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

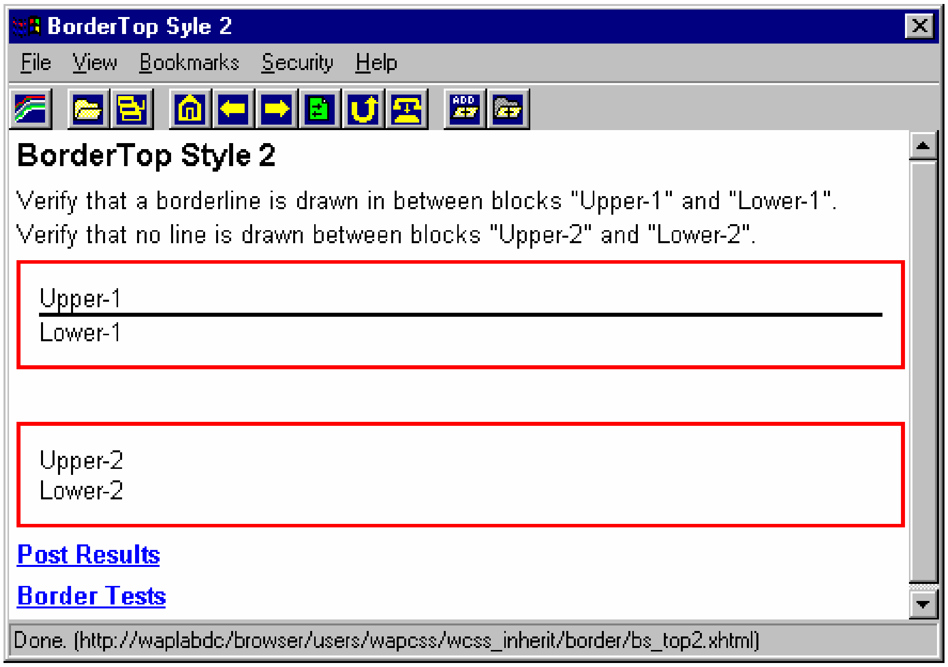
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1". Verify that no line is drawn between blocks "Upper-2" and "Lower-2".



BorderTop Style 2

Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1".
Verify that no line is drawn between blocks "Upper-2" and "Lower-2".

Upper-1
Lower-1

Upper-2
Lower-2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_top2.xhtml)

45.29.31 BorderBottom Style 1 wcss_inherit/border/bs_bott1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

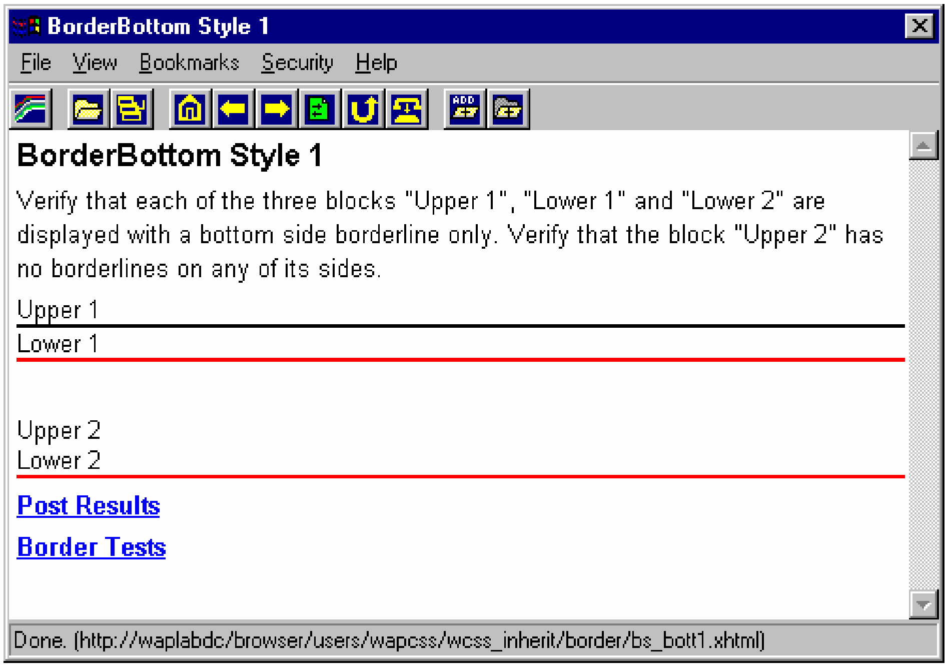
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that each of the three blocks "Upper 1", "Lower 1" and "Lower 2" are displayed with a bottom side borderline only. Verify that the block " Upper 2" has no borderlines on any of its sides.



BorderBottom Style 1

Verify that each of the three blocks "Upper 1", "Lower 1" and "Lower 2" are displayed with a bottom side borderline only. Verify that the block "Upper 2" has no borderlines on any of its sides.

Upper 1

Lower 1

Upper 2

Lower 2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_bott1.xhtml)

45.29.32 BorderBottom Style 2 wcss_inherit/border/bs_Bott2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

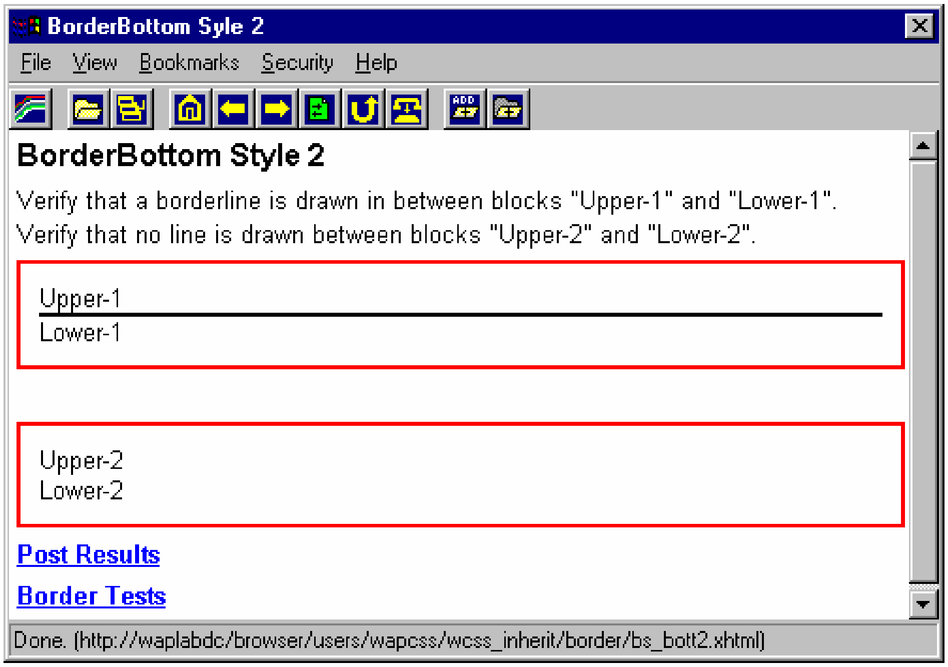
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1". Verify that no line is drawn between blocks "Upper-2" and "Lower-2".



BorderBottom Style 2

Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1".
Verify that no line is drawn between blocks "Upper-2" and "Lower-2".

Upper-1
Lower-1

Upper-2
Lower-2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_bott2.xhtml)

45.29.33 BorderLeft Style 1 wcss_inherit/border/bs_left1.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

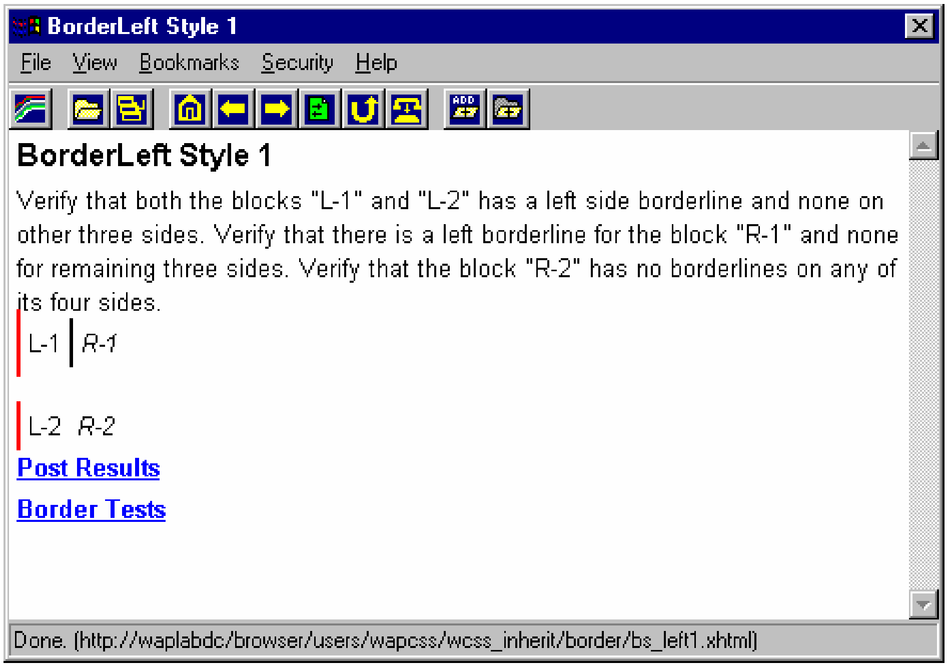
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that both the blocks "L-1" and "L-2" has a left side borderline and none on other three sides. Verify that there is a left borderline for the block "R-1" and none for remaining three sides. Verify that the block "R-2" has no borderlines on any of its four sides.



BorderLeft Style 1

Verify that both the blocks "L-1" and "L-2" has a left side borderline and none on other three sides. Verify that there is a left borderline for the block "R-1" and none for remaining three sides. Verify that the block "R-2" has no borderlines on any of its four sides.

L-1 *R-1*

L-2 *R-2*

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_left1.xhtml)

45.29.34 BorderLeft Style 2 **wcss_inherit/border/bs_left2.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

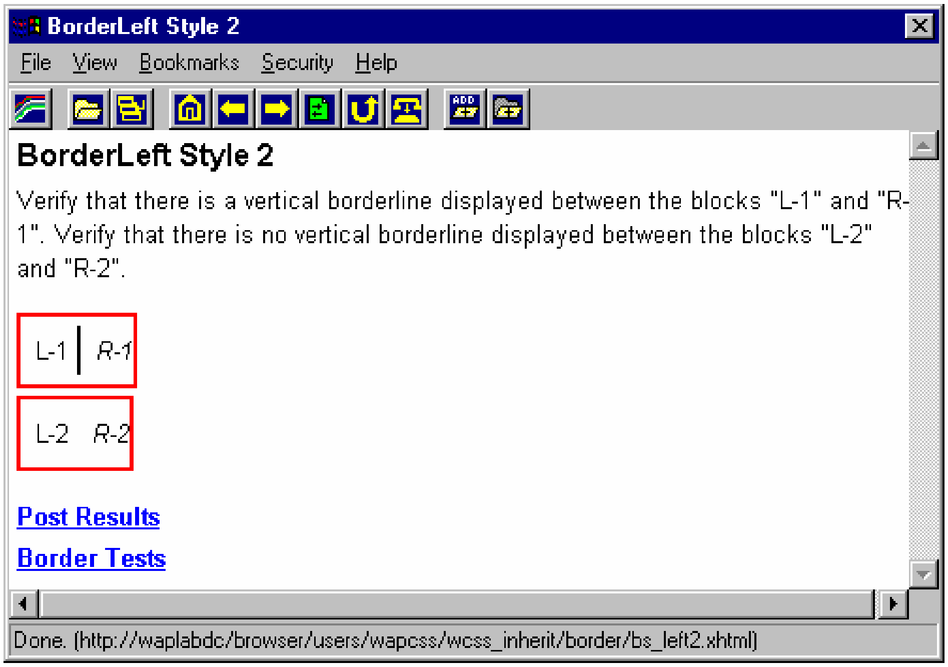
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".



BorderLeft Style 2

Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_left2.xhtml)

45.29.35 **BorderRight Style 1** **wcss_inherit/border/bs_right1.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

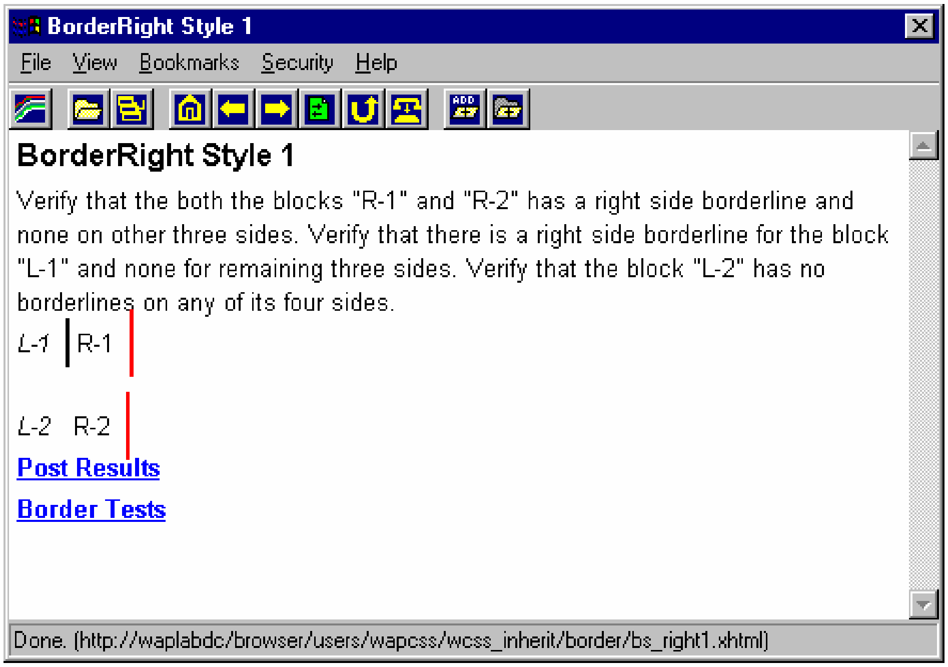
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the both the blocks "R-1" and "R-2" has a right side borderline and none on other three sides. Verify that there is a right side borderline for the block "L-1" and none for remaining three sides. Verify that the block "L-2" has no borderlines on any of its four sides.



BorderRight Style 1

Verify that the both the blocks "R-1" and "R-2" has a right side borderline and none on other three sides. Verify that there is a right side borderline for the block "L-1" and none for remaining three sides. Verify that the block "L-2" has no borderlines on any of its four sides.

L-1 R-1

L-2 R-2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_right1.xhtml)

45.29.36 BorderRight Style 2 wcss_inherit/border/bs_right2.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

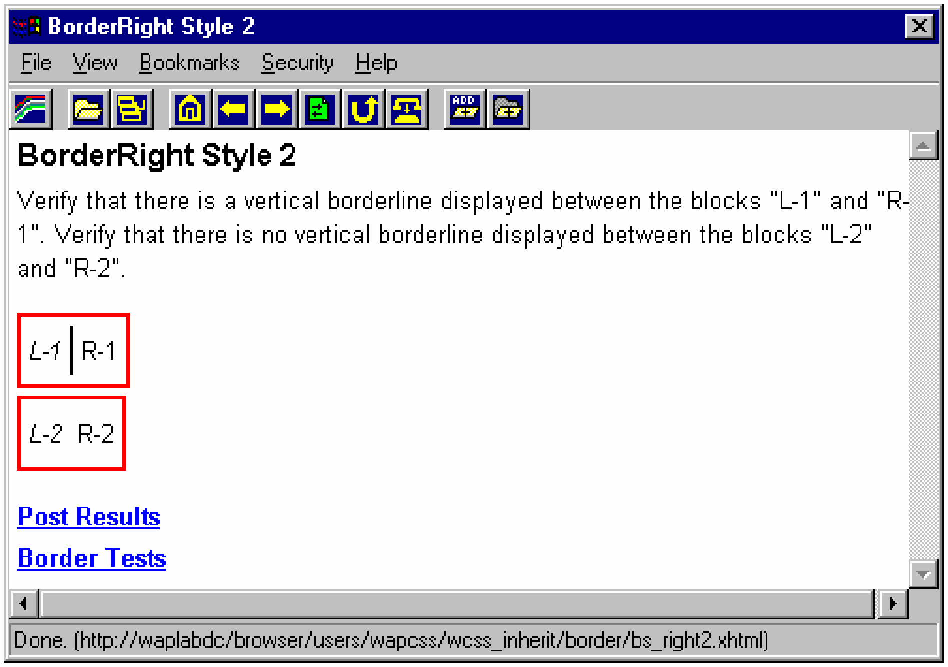
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".



BorderRight Style 2

Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_right2.xhtml)

45.29.37 Border Style wcss_inherit/border/bs_all.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

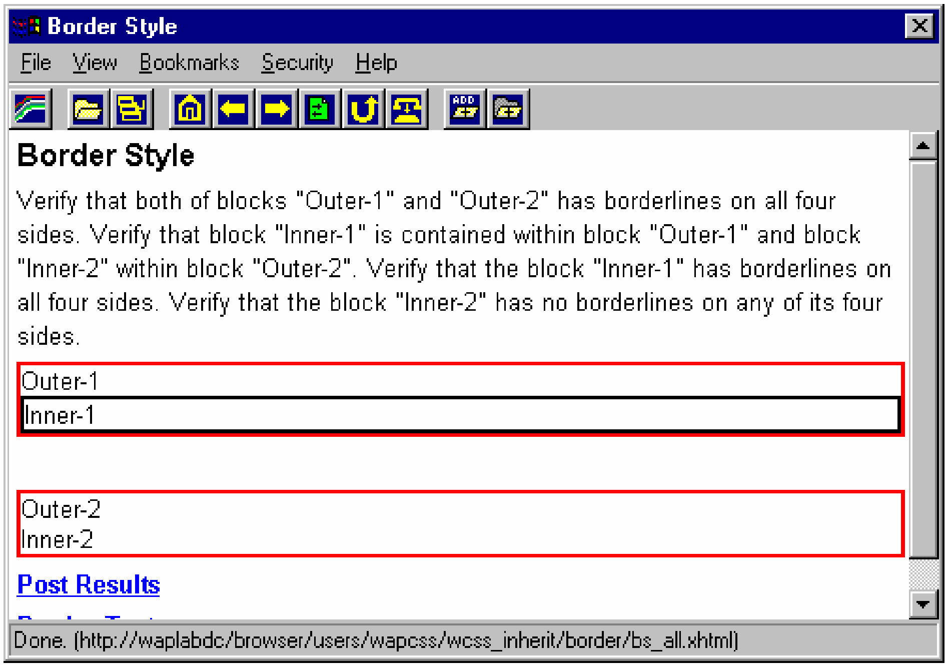
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that both of blocks "Outer-1" and "Outer-2" has borderlines on all four sides. Verify that block "Inner-1" is contained within block "Outer-1" and block "Inner-2" within block "Outer-2". Verify that the block "Inner-1" has borderlines on all four sides. Verify that the block "Inner-2" has no borderlines on any of its four sides.



Border Style

Verify that both of blocks "Outer-1" and "Outer-2" has borderlines on all four sides. Verify that block "Inner-1" is contained within block "Outer-1" and block "Inner-2" within block "Outer-2". Verify that the block "Inner-1" has borderlines on all four sides. Verify that the block "Inner-2" has no borderlines on any of its four sides.

Outer-1
Inner-1

Outer-2
Inner-2

Post Results

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_all.xhtml)

19.6 Border Shorthand Properties

45.29.38 BorderTop Shorthand wcss_inherit/border/bsh_top.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

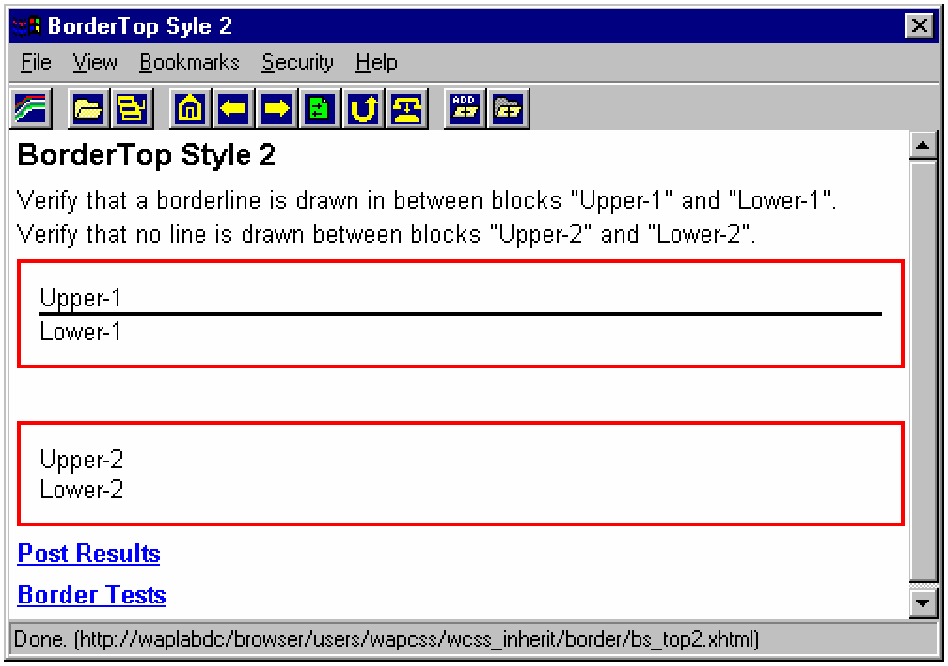
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a borderline drawn between blocks "Upper-1" and "Lower-1". Verify that no borderline is drawn between blocks "Upper-2" and "Lower-2".



BorderTop Style 2

Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1".
Verify that no line is drawn between blocks "Upper-2" and "Lower-2".

Upper-1
Lower-1

Upper-2
Lower-2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_top2.xhtml)

45.29.39 BorderBottom Shorthand wcss_inherit/border/bsh_bott.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

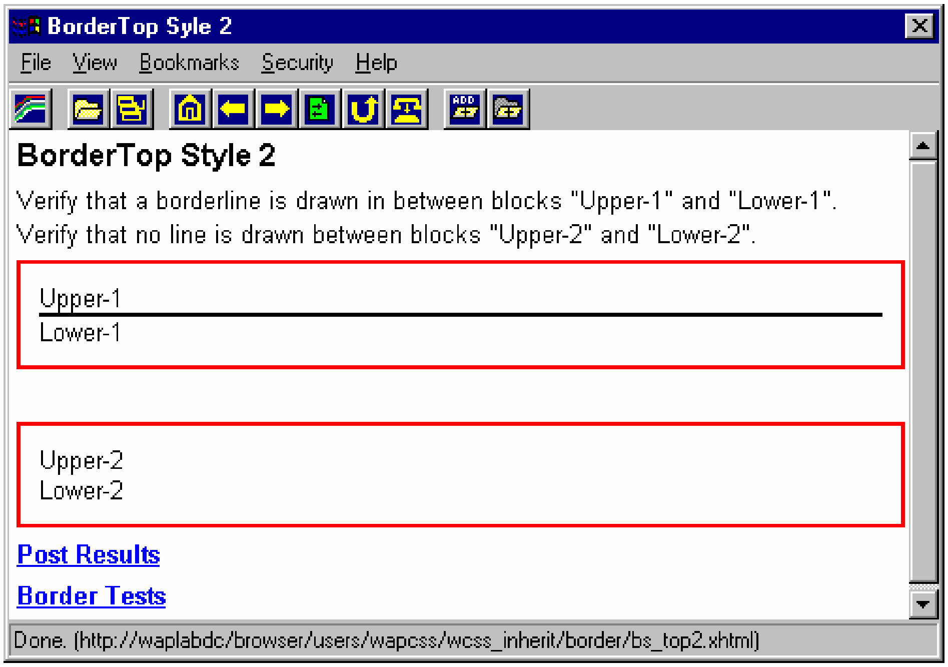
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a borderline drawn between blocks "Upper-1" and "Lower-1". Verify that no borderline is drawn between blocks "Upper-2" and "Lower-2".



BorderTop Style 2

Verify that a borderline is drawn in between blocks "Upper-1" and "Lower-1".
Verify that no line is drawn between blocks "Upper-2" and "Lower-2".

Upper-1
Lower-1

Upper-2
Lower-2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_top2.xhtml)

45.29.40 BorderLeft Shorthand wcss_inherit/border/bsh_left.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

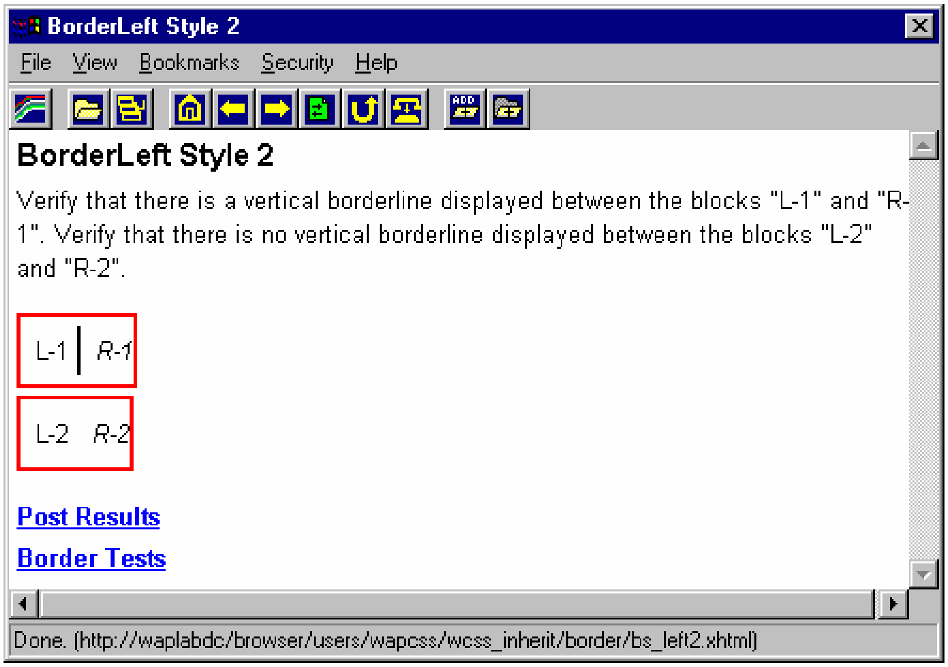
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical borderline drawn between inline blocks "L-1" and "R-1". Verify that there is no vertical borderline drawn between inline blocks "L-2" and "R-2".



BorderLeft Style 2

Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_left2.xhtml)

45.29.41 BorderRight Shorthand wcss_inherit/border/bsh_right.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

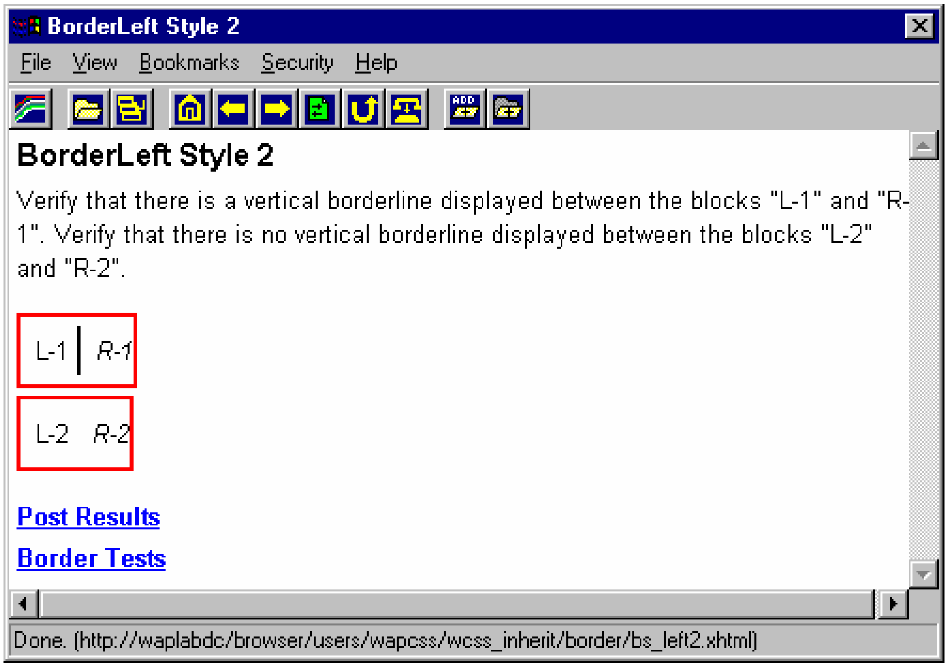
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that there is a vertical borderline drawn between inline blocks "L-1" and "R-1". Verify that there is no vertical borderline drawn between inline blocks "L-2" and "R-2".



BorderLeft Style 2

Verify that there is a vertical borderline displayed between the blocks "L-1" and "R-1". Verify that there is no vertical borderline displayed between the blocks "L-2" and "R-2".

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/bs_left2.xhtml)

45.29.42 Border Shorthand wcss_inherit/border/border_sh.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

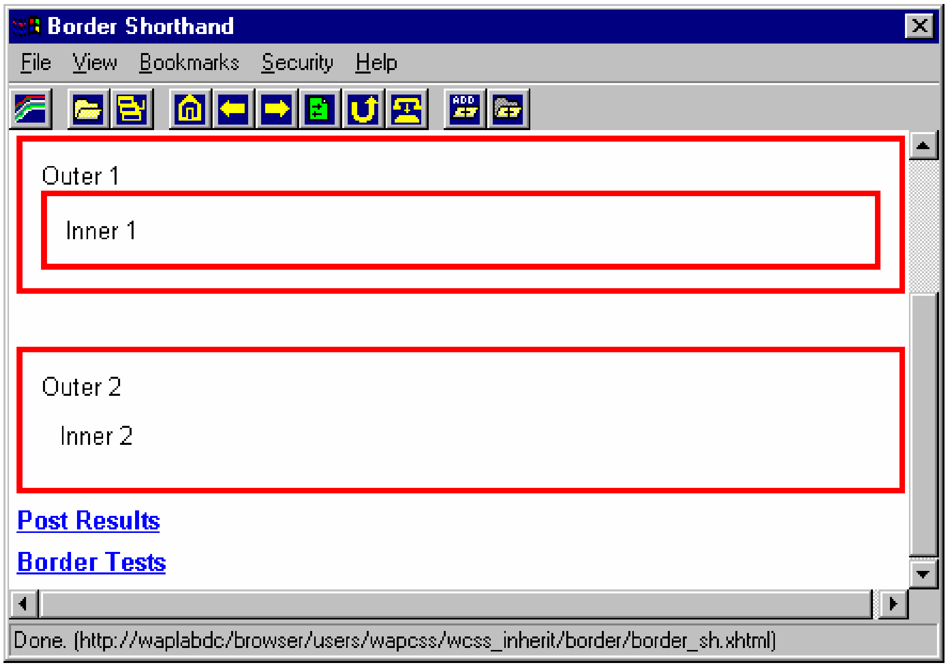
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that both of blocks "Outer-1" and "Outer-2" has thick and red colored borderlines on all four sides. Verify that block "Inner-1" is contained within block "Outer-1" and block "Inner-2" within block "Outer-2". Verify that the block "Inner-1" has also thick and red colored borderlines on all four sides. Verify that the block "Inner-2" has no borderlines on any of its four sides.



Outer 1

Inner 1

Outer 2

Inner 2

Post Results

Border Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/border/border_sh.xhtml)

19.7 Colors

45.29.43 Foreground Color wcss_inherit/colors/fgcolor.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that text of both "Para-1" and "Para-2" is displayed red color font.

45.29.44 Background Color wcss_inherit/colors/bgcolor.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that "Para-1" is displayed with red colored back-ground, "Para-2' with yellow colored back-ground and "Para-3' with red colored back-ground.

45.29.45 Background Image wcss_inherit/colors/bgimge.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that this page loads without crashing the browser.

19.8 Fonts

45.29.46 Font-Style wcss_inherit/fonts/font_style.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Line-1" and "Line-2" is displayed in italic font style.

45.29.47 Font-variant wcss_inherit/fonts/font_variant.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Line-1" and "Line-2" is displayed in small-caps font.

45.29.48 Font-Weight **wcss_inherit/fonts/font_weight.xhtml**

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Line-1" and "Line-2" is displayed in equally dark font.

45.29.49 Font-Size **wcss_inherit/fonts/font_size.xhtml**

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Line-1" and "Line-2" is displayed in font of equally big size.

45.29.50 Font Shorthand **wcss_inherit/fonts/font_sh.xhtml**

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Line-1" and "Line-2" is displayed in uppercase, italic, bold and large size fonts.

19.9 Lists

45.29.51 List-style-type wcss_inherit/lists/lstype.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the list-items of both "List-1" and "List-2" are displayed with square shaped markers.

45.29.52 List-style-position wcss_inherit/lists/lsposition.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that list-items of both "List 1" and "List 2" are displayed with markers placed inside the list-item.

45.29.53 List-style-image wcss_inherit/lists/lsimage.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that an image is displayed as marker for each list-item of both "List 1" and "List 2".

45.29.54 List-style Short-hand wcss_inherit/lists/ls_sh.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that list-items of both the "List 1" and "List 2" are displayed with square shaped bullets placed in in-side position.

19.10 Text

45.29.55 Text-Indentation wcss_inherit/text/text_indent.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that in each of the "Para-1" and "Para-2", the first-line text is indented further than other lines.

45.29.56 Text-Align wcss_inherit/text/text_align.xhtml

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the content of both the "Para-1" and "Para-2" is right aligned.

45.29.57 Text-Decoration **wcss_inherit/text/text_decorate.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in each of the "Para-1" to "Para-4" is displayed underlined.

45.29.58 Text-Transformation **wcss_inherit/text/text_transform.xhtml**

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Para-1" and "Para-2" is displayed in all upper case letters.

45.29.59 Text-White Space **wcss_inherit/text/text_ws.xhtml**

Assertion 1: 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 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the text in both "Para-1" and "Para-2" is displayed with large intra-word space intact.

19.11 Visual Effects

45.29.60 **Visibility** **wcss_inherit/vEffects/ve_visibility.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

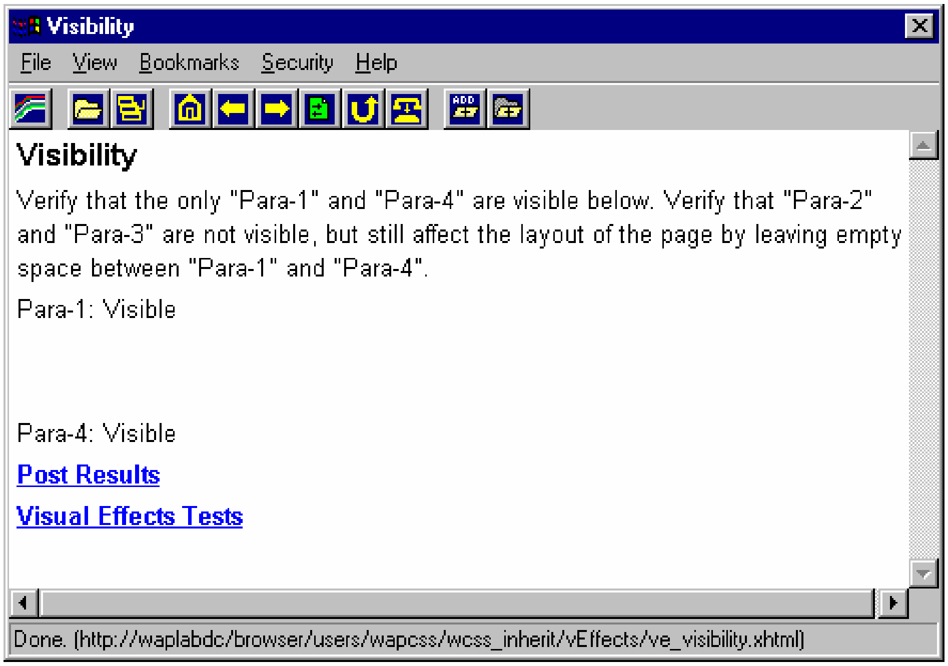
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the only "Para-1" and "Para-4" are visible. Verify that "Para-2" and "Para-3" are not visible, but still affect the layout of the page by leaving empty space between "Para-1" and "Para-4".



Visibility

Verify that the only "Para-1" and "Para-4" are visible below. Verify that "Para-2" and "Para-3" are not visible, but still affect the layout of the page by leaving empty space between "Para-1" and "Para-4".

Para-1: Visible

Para-4: Visible

Post Results

Visual Effects Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/vEffects/ve_visibility.xhtml)

19.12 Visual Formatting

45.29.61 Display wcss_inherit/vFormat/vf_display.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the blocks "B1" to "B4" are displayed in same line. Verify that the block "B5" is displayed in separate line.

45.29.62 Float wcss_inherit/vFormat/vf_float.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

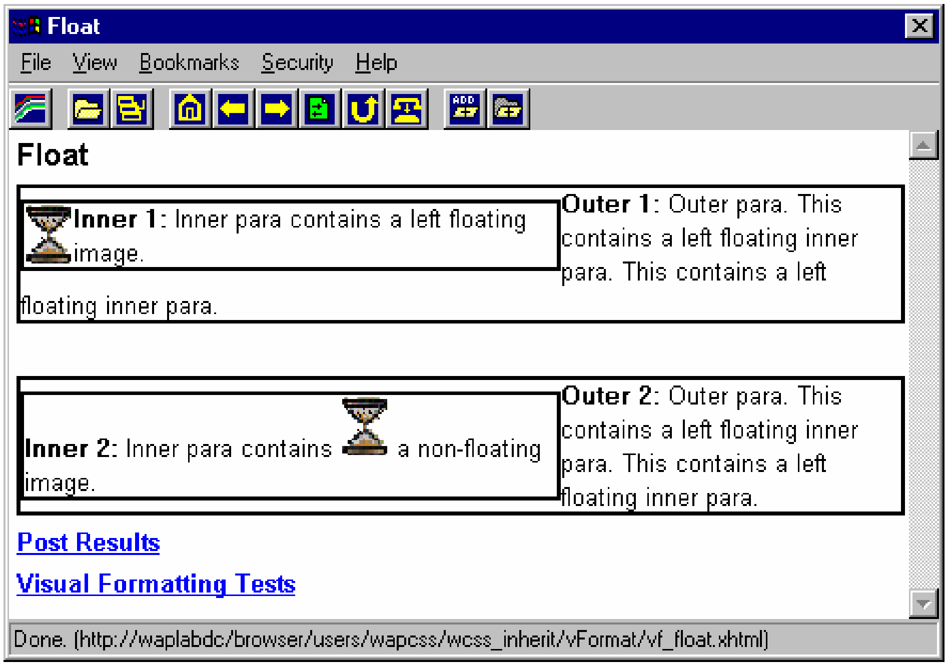
Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that the para “Inner-1” floats to the left of para “Outer-1”. Verify that the image in para “Inner-1” floats to the left of the para “Inner-1”. Verify that the para “Inner-2” floats to the left of para “Outer-2”. Verify that the image in para “Inner-2” does not float to the left of the para “Inner-1”.

Example of display:



Float

Inner 1: Inner para contains a left floating image.

Outer 1: Outer para. This contains a left floating inner para. This contains a left floating inner para.

Inner 2: Inner para contains a non-floating image.

Outer 2: Outer para. This contains a left floating inner para. This contains a left floating inner para.

Post Results

Visual Formatting Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/vFormat/vf_float.xhtml)

45.29.63 Clear wcss_inherit/vFormat/vf_clear.xhtml

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2

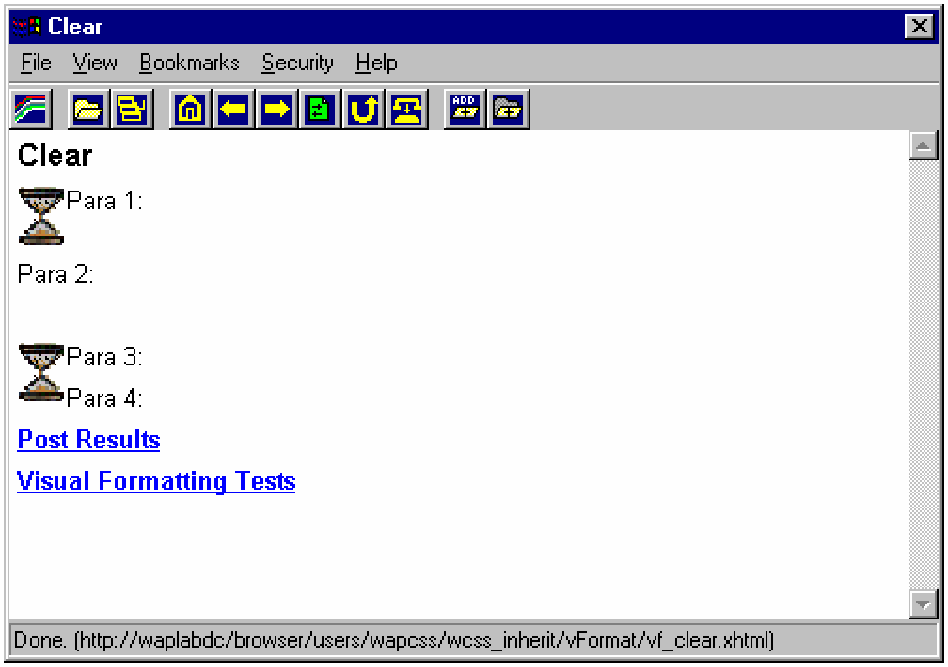
Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that Para-2 clears the first image and gets displayed below it. Verify that Para-4 does not clear the second image and gets displayed to right of it.

Example of display:



Clear

Para 1:

Para 2:

Para 3:

Para 4:

Post Results

Visual Formatting Tests

Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/vFormat/vf_clear.xhtml)

45.29.64 Width **wcss_inherit/vFormat/vf_width.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

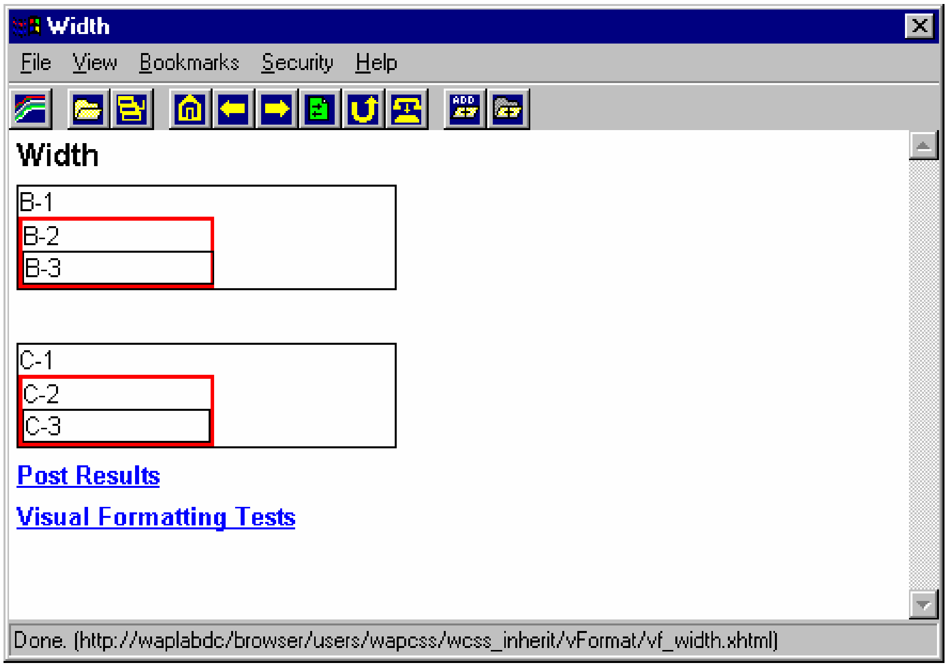
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that block “B-2” is displayed within block “B-1” and has 50% of the width of block “B-1”. Verify that block “B-3” is displayed within block “B-2” and has same width as block “B-2” (inherits computed value of B-2 width). Verify that block “C-2” is displayed within block “C-1” and has 50% of the width of block “C-1”. Verify that block “C-3” is displayed within block “C-2” and has same width as block “C-2” (gets same width by default value ‘auto’).



Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/vFormat/vf_width.xhtml)

45.29.65 Height **wcss_inherit/vFormat/vf_height.xhtml**

Assertion 1: When a property defined as not inherited is set in an element, all the children elements, which do not overwrite the property, must use the initial value defined for the property.

Traceability:

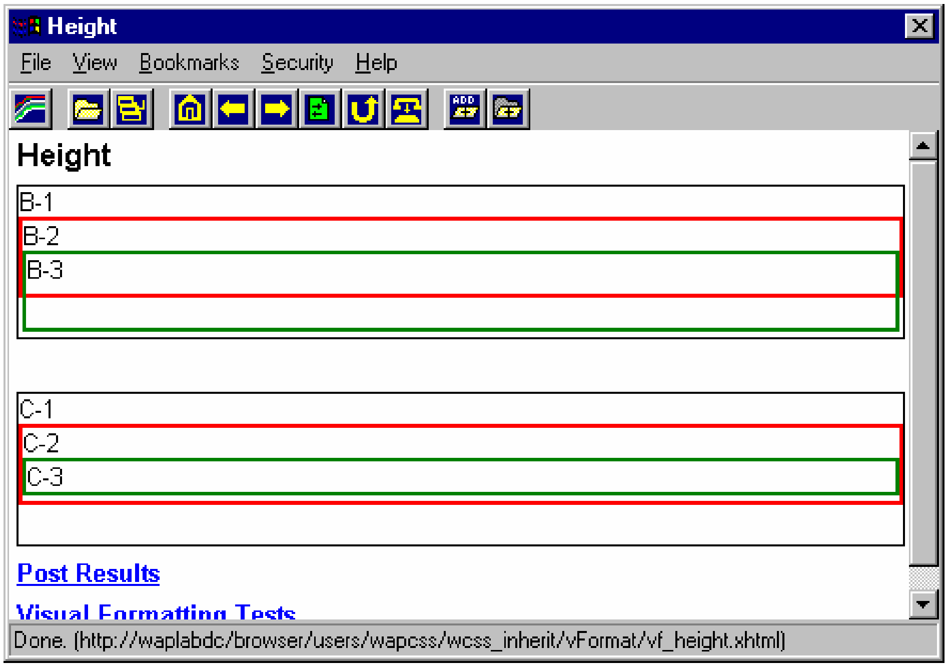
[WAPCSS]	Section 7
[CSS2]	Section 6.2

Assertion 2: When a property is set explicitly as 'inherit' in an element, the property of the element must have the same computed value as its parent element.

Traceability:

[WAPCSS]	Section 7
[CSS2]	Section 6.2.1

Verification: Verify that block “B-2” is displayed within block “B-1” and has 50% of the height of block “B-1”. Verify that block “B-3” is displayed with same height as block “B-2” (inherits computed value of B-2 height). Verify that block “C-2” is displayed within block “C-1” and has 50% of the height of block “C-1”. Verify that block “C-3” is displayed within block “C-2” (gets some height by default value ‘auto’).



Done. (http://waplabdc/browser/users/wapcss/wcss_inherit/vFormat/vf_height.xhtml)

20 TABLE TESTS

This section contains test cases for testing different [WAPCSS] styles assigned to tables and its elements.

45.30.01 **Table 1** **wcss_tables/table1.xhtml**

Assertion: When text-align and border style properties are assigned to table elements, UA must render the table with the assigned styles.

Traceability:

[CSS2]	Section 17.4

Verification: Verify that a table is displayed with a caption. Verify that the caption is left aligned and in italic font. Verify that entire table has thick borders. Verify that the table has three rows of three cells each. Verify that the individual cells have red borders on top and bottom sides and blue borders on left and right sides.

45.30.02 **Table 2** **wcss_tables/table2.xhtml**

Assertion: When vertical-alignment style properties are assigned to table elements, UA must render the table with the assigned styles.

Traceability:

[CSS2]	Section 17.4

Verification: Verify the vertical alignment of the contents of cells is: first row - baseline aligned, second row - top aligned, third row - bottom aligned and fourth row - middle aligned.

45.30.03 **Table 3** **wcss_tables/table3.xhtml**

Assertion: When any style properties are assigned to an empty table, then UA must not display any tables.

Traceability:

[CSS2]	Section 17.4

Verification: Verify that no tables are displayed in the page.

45.30.04 **Table 4** **wcss_tables/table4.xhtml**

Assertion: When text-align style properties are assigned to table elements, UA must render the table with the assigned styles.

Traceability:

[CSS2]	Section 17.4

Verification: Verify the horizontal alignment of the contents of cells is: first row - default aligned, second row - right aligned, third row - center aligned and fourth row - left aligned.

45.30.05 **Table 5** **wcss_tables/table5.xhtml**

Assertion: When vertical-alignment and text-align style properties are assigned to table elements, UA must render the table with the assigned styles.

Traceability:

[CSS2]	Section 17.4

Verification: Verify that both the horizontal and vertical alignment of the contents of cells of the table is as described in the cells.

45.30.06 **Table 6** **wcss_tables/table6.xhtml**

Assertion: The background-color set on any table element will only be visible, if the layers above it have a transparent background.

Traceability:

[CSS2]	Section 17.4

Verification: Verify that a table is displayed and each of its cells have background color displayed as described.

21 CSS2 EXTENSIONS

A [WAPCSS] supporting user-agent may also support additional CSS2 features. This means, when we load a CSS2 conforming style sheet which is not part of [WAPCSS], the user-agent may choose to render these styles. This section contains some of such CSS2 test cases (45.30.01 to 45.30.09). Even if the user-agent cannot support these 'CSS2 only' styles, it is expected to ignore the unrecognized styles and display the page without crashing.

21.1 Selectors

45.31.01 **Child Selector** **wcss_css2_ext/sel_child.xhtml**

Assertion: When the Child Selector of the form " E > F" is used, the rule must be applied to every instance of the F element, which is a direct child of element E.

Traceability:

[CSS2]	Section 5.6

Verification: Verify that the entire text of "Para-2" is displayed as normal text. If the user-agent supports the child selector, then verify that the entire text of "Para-1" is displayed in bold letters only.

45.31.02 **Adjacent Siblings** **wcss_css2_ext/sel_sibling.xhtml**

Assertion: When adjacent sibling selectors of the form " E + F" is used, the rule must be applied to every instance of the F element, which directly follows element E and when both E and F share the same parent.

Traceability:

[CSS2]	Section 5.7

Verification: If the user-agent supports the adjacent sibling selectors, then verify that only the word BOSTON appears in italic in the page.

45.31.03 Attribute Selectors 1 wcss_css2_ext/sel_attributes1.xhtml

Assertion 1: When attribute selectors of the form " E [att] " is used, the rule must be applied to every instance of the E element, which has the attribute "att" set, what ever is the value of the attribute.

Traceability:

[CSS2]	Section 5.8.1

Assertion 2: When attribute selectors of the form " E [att=val] " is used, the rule must be applied to every instance of the E element, which has the attribute "att" set exactly to the value "val".

Traceability:

[CSS2]	Section 5.8.1

Verification: If the user-agent supports the attribute selectors of the form “E[att]” and “E [att=val]” , then verify that only the word BOSTON appears in all italic in the page.

45.31.04 Attribute Selectors 2 wcss_css2_ext/sel_attributes2.xhtml

Assertion: When attribute selectors of the form " E [att~=val] " is used, the rule must be applied to every instance of the E element which has the attribute "att" set to a space separated list of words and one of which is exactly "val".

Traceability:

[CSS2]	Section 5.8.1

Verification: If the user-agent supports the attribute selectors of the form E[att~=val], then verify that only the word BOSTON appears in all italic in the page.

45.31.05 Attribute Selectors 3 wcss_css2_ext/sel_attributes3.xhtml

Assertion: When attribute selectors of the form " E [att | =val] " is used, the rule must be applied to every instance of the E element which has the attribute "att" set to a hyphen separated list of words beginning with "val".

Traceability:

[CSS2]	Section 5.8.1

Verification: If the user-agent supports the attribute selectors of the form E[att | = val], then verify that only the word BOSTON appears in all italic in the page.

45.31.06 **Attribute Defaults** **wcss_css2_ext/sel_att_default.xhtml**

Assertion: When attribute selectors of the form " E [att=val] " is used and if "val" is defined as the default value for the attribute "att" for element E in the document's DTD, then the rule must be applied to every instance of the A element in the document where ever the attribute 'att' is not set.

Traceability:

[CSS2]	Section 5.8.2

Verification: If the user-agent supports the attribute selector with default value of the form “E [att = default]”, then verify that only the word BOSTON appears in italic in the page.

45.31.07 **First Child** **wcss_css2_ext/sel_firstchild.xhtml**

Assertion: When the first-child pseudo class selector of the form "E**:**first-child " is used, then the rule must be applied to every instance of the E element in the document where ever it is the first child of some other element.

Traceability:

[CSS2]	Section 5.11.1

Verification: If the user-agent supports the “first child” selector, then verify that only the word BOSTON appears in all italic text .

45.31.08 **First Line** **wcss_css2_ext/sel_firstline.xhtml**

Assertion : When the first-line pseudo element selector of the form "p**:**first-line " is used, then the rule must be applied to the first formatted line of every paragraph in the document.

Traceability:

[CSS2]	Section 5.11.1

Verification: If the user-agent supports the “first line” selector, then verify that only the first line of the "Para-1" is displayed in all italic characters and the rest of the lines are displayed in normal case.

45.31.09 **First Letter** **wcss_css2_ext/sel_firstletter.xhtml**

Assertion : When the first-letter pseudo element selector of the form "p**u**ffiest-letter " is used, then the rule must be applied to the first formatted letter of every paragraph in the document.

Traceability:

[CSS2]	Section 5.11.1

Verification: If the user-agent supports the “first letter” selector, then verify that the first character of the "Para-1" is displayed in bold and italic.

22 CSS MP EXTENSIONS

The only additional requirements of CSS Mobile Profile [CSSMP] over [WAPCSS] are the support of the child selector and the following length data types.

- in (inches)
- cm (centimeters)
- mm (millimeters)
- pt (points - one point equals 1/72h of an inch)
- pc (picas – one pica equal to 12 points, i.e., 1/6th of an inch).

This section covers the test cases (45.32.01 to 45.32.08) for testing these length units and the child selector. The [WAPCSS] supporting user-agent should load all these tests without error, but may choose to ignore these [CSSMP] only' data types and selector.

45.32.01 Inch Units wcss_cssmp_ext/dt_in.xhtml

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

Traceability:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that both “Block-1” and “Block-2” are displayed with borders. If the user-agent supports length unit 'in', then verify that the borders of "Block-2" are rendered thicker than the borders of "Block-1"

45.32.02 mm Units wcss_cssmp_ext/dt_mm.xhtml

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

Traceability:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that both “Block-1” and “Block-2” are displayed with borders. If the user-agent supports length unit 'mm', then verify that the borders of "Block-2" are rendered thicker than the borders of "Block-1"

45.32.03 **cm Units** **wcss_cssmp_ext/dt_cm.xhtml**

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

Traceability:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that both “Block-1” and “Block-2” are displayed with borders. If the user-agent supports the length unit 'cm', then verify that the borders of "Block-2" are rendered thicker than the borders of "Block-1"

45.32.04 **pt Units** **wcss_cssmp_ext/dt_pt.xhtml**

Assertion : Verify that the page loads without any errors. When a property that contains a 'pt' length value is encountered, the user agent must support that property value.

Traceability:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that both “Block-1” and “Block-2” are displayed with borders. If the user-agent supports the length unit 'pt', then verify that the borders of "Block-2" are rendered thicker than the borders of "Block-1"

45.32.05 **pc Units** **wcss_cssmp_ext/dt_pc.xhtml**

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

Traceability:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that both “Block-1” and “Block-2” are displayed with borders. If the user-agent supports the length unit 'pc', then verify that the borders of "Block-2" are rendered thicker than the borders of "Block-1"

45.32.06 **Units 1** **wcss_cssmp_ext/dt_mmcmin.xhtml**

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

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

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

Traceability of above three assertions:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that each of the “Block-1” to “Block-3” is displayed with borders. If the user-agent supports the length units 'mm', 'cm' and 'in', then verify that the borders of "Block-3" are rendered thickest, borders of "Block-1" thinnest and the borders of "Block-2" in-between these thicknesses.

45.32.07 **Units 2** **wcss_cssmp_ext/dt_ptpcin.xhtml**

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

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

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

Traceability of above three assertions:

[CSSMP]	Section 5.2
[CSS2]	Section 4.3.2

Verification: Verify that each of the “Block-1” to “Block-3” is displayed with borders. If the user-agent supports length units 'pt', 'pc' and 'in, then verify that the borders of "Block-3" are rendered thickest, borders of "Block-1" thinnest and the borders of "Block-2" in-between these thicknesses.

45.32.08 **Child Selector** **wcss_cssmp_ext/mpsel_child.xhtml**

Assertion: When the Child Selector of the form " E > F" is used, the rule must be applied to every instance of the F element, which is a direct child of element E.

Traceability:

[CSSMP]	Section 3
[CSS2]	Section 5.6

Verification: Verify that the entire text of "Para-2" is displayed as normal text. If the user-agent supports the child selector, then verify that the entire text of "Para-1" is displayed only in italic characters.