Enabler Test Report Browsing v2.x

OMA TestFest (May 2005)
Version 27-May-2005

Open Mobile Alliance
OMA-Enabler_Test_Report-BROWSING-2x-20050527

Contents

1. Scope

This report describes the results from the testing carried out at OMA TestFest9 May 2005 concerning the following Browsing related specifications:

- XHTML Mobile Profile 1.1 [XHTMLMP]

- Wireless CSS Specification Version 1.1 [WCSS]

- Wireless Markup Language Version 1.3 [WML1]

- ECMAScript Mobile Profile Version 1.0 [ESMP]

- User Agent Profile Version 2.0 [UAProf]

2. References

2.1 Normative References

[OMAIOPPROC]	OMA Interoperability Policy and Process, http://www.openmobilealliance.org/
[XHTMLMP]	"XHTML Mobile Profile 1.1". OMA-WAP-XHTMLMP-V1_1-20020904-C
	http://www.openmobilealliance.org/
[WCSS]	"Wireless CSS Specification Version 1.1",
	http://www.openmobilealliance.org/
[WML1]	"Wireless Markup Language Version 1.3", Open Mobile Alliance™. WAP-191-WML. URL:http://www.openmobilealliance.org/
[ESMP]	"WAP - ECMAScript Specification". Open Mobile Alliance™. OMA-ESMP-V1_0_0-20031120-C. URL: http://www.openmobilealliance.org/release_program/docs/CopyrightClick.asp?pck=Browsing&file=V2_2_C/OMA-WAP-ESMP-v1_0-20031120-C.pdf
[UAProf]	"User Agent Profile", Open Mobile Alliance, OMA-WAP-UAProf-v2_0-20030503-C URL: http://www.openmobilealliance.org/
[EPTR]	Enabler Product Test Report
[ETP]	Enabler Test Plan
[XHML_ETS]	Enabler Test Specification for XHTML 1.1 Approved Version 1.1, 18th November 2004
[WCSS_ETS]	Enabler Test Specification for WCSS 1.1 Approved Version 1.1, 27th October 2004
[WML1_ETS]	Enabler Test Specification for WML 1.3 Approved Version 1.3, 18th Novemer 2004
[ESMP_ETS]	Enabler Test Specification for ESMP 1.0 Approved Version 1.0, 18th Novemer 2004
[UAPROF_ETS]	Enabler Test Specification for UAPROF 2.0 Approved Version 1.0, 18th November 2004

2.2 Informative References

3. Terminology and Conventions

3.1 Conventions

This is an informative document, i.e. the document does not intend to contain normative statements.

3.2 Definitions

None.

3.3 Abbreviations

ECMA	European Computers Manufacturers Association
ESMP	ECMAScript Mobile Profile
ETR	Enabler Test Requirements
ETS	Enabler Test Specification
SCR	Static Conformance Statement
UAProf	User Agent profile
WCSS	Wireless Cascading Style Sheets
WML	Wireless Markup Language
XHTML	Extensible HyperText Markup Language

4. Summary

This report gives details of the testing carried out during the OMA TestFest9 (May 2005) for the Browsing enablers.

The report is compiled on behalf of OMA by NCC Group.

The work and reporting has followed the OMA IOP processes and policies [OMAIOPPROC].

5. Test Details

5.1   Documentation

This chapter lists the details of the enabler and any documentation, tools or test suites used to prove the enabler.

Date:	May 2005
Location:	Helsinki, Finland
Enabler:	Browsing v2.x
Process:	OMA Interoperability Policy and Process [OMAIOPPROC]
Type of Testing	Interoperability Testing
Products tested:	Client-to-server
Test Plan:	-
Test Specification:	OMA-ETS-XHTML-V1_1-20041118-A.doc: Enabler Test Specification for XHTML 1.1 OMC-ETS-WCSS-V1_1-20041027-D.doc: Enabler Test Specification for WCSS 1.1 OMC-ETS-WML-1_3-20041118-A.doc: Enabler Test Specification for WML 1.3 OMA-ETS-ESMP-V1_0-20041118-A.doc: Enabler Test Specification for ESMP 1.0 OMA-ETS-UAPROF-V2_0-20041118-A.doc: Enabler Test Specification for UAPROF 2.0
Test Tool:	None
Test Code:	None
Type of Test event:	TestFest
Participants:	MobileSoft Technology (Nanjing) Co., Ltd., Nokia, Openwave Systems and IXI Mobile Inc
Number of Client Products:	4
Participating Technology Providers for clients:	MobileSoft Technology (Nanjing) Co., Ltd., Nokia, Openwave Systems and IXI Mobile Inc
Number of Server Products:	None
Participating Technology Providers for servers:	n/a
Number of test sessions completed:	4 of 4

5.2 Test Case Statistics

5.2.1 Test Case Summary

This chapter gives an overview of the result for all test cases included in [ETS].

The following status is used in the tables below:

- **Total number of TCs**: Used in the summary to indicate how many test cases there are in total.

- **Number of passed:** Used in the summary to indicate how many of the total test cases successfully passed.

- **Number of failed:** Used in the summary to indicate how many of the total test cases failed.

- **Number of N/A:** Used in the summary to indicate how many of the total test cases have not been run due to one of the implementations not supporting the functionality required to run this test case.

- **Number of OT:** Used in the summary to indicate how many of the total test cases have not been run due to no time to run the test case.

- **Number of INC:** Used in the summary to indicate how many of the total test cases have not been run due to functionality not being tested due to an error in the implementation or other functionality that is required to run this test case.

Test Section:	Number of test sessions:	Total number of TCs:	Number of Passed:	Number of Failed:	Number of N/A:	Number of OT:	Number of INC:	Total:
xHTML v1.1 (-int-)	4	11	29	5	0	0	10	44
xHTML v1.1 (-con-)	4	18	36	15	0	0	21	72
WCSS v1.1 (-int-)	4	23	56	30	2	0	4	92
WCSS v1.1 (-con-)	4	36	63	61	6	1	13	144
WML 1.3	4	201	627	118	5	36	18	804
ESMP v1.0 (-int-)	4	16	0	0	64	0	0	64
ESMP v1.0 (-con-)	4	58	0	0	232	0	0	232
UAProf 2.0 (-int-)	4	8	0	0	32	0	0	32
UAProf 2.0 (-con-)	4	3	0	0	12	0	0	12
Total	**4**	**374**	**811**	**229**	**353**	**37**	**66**	**1496**

5.2.2   Test Case List

This chapter lists the statistics for all test cases included in [ETS].

The following status is used in the tables below:

- **No. of runs(R):** Used to indicate how many times the test cases have been run in total.

- **No. of passed(P):** Used to indicate how many times the test case has been run with successful result.

- **No. of failed(F):** Used to indicate how many times the test case has been run with failed result

- **No. of OT(O):** Used to indicate how many times the test case has not been run due to no time available.

- **No. of INC(I):** Used to indicate how many times the test case has not been run due to errors being found in other functionality required for running this test case.

- **PR:** Used to indicate if any PRs (Problem Reports) have been issued during testing.

- **Note:** Used to indicate the cause of Inconclusive or Fail verdicts.

5.2.2.1   XHTML v1.1

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
xHTML-1.1-int-1	To test basic form Module. Test input element with type set to text but with the illegal attribute checked set	4	3	0	0	1	-	
xHTML-1.1-int-2	To test optgroup element in Forms module	4	2	2	0	0	-	
xHTML-1.1-int-3	To test Basic Tables	4	2	1	0	1	-	
xHTML-1.1-int-4	To test hr element in Presentation module	4	3	1	0	0	-	
xHTML-1.1-int-5	To test base Module. The base element specifies a base URL. The purpose is to check that this base URL is used in a correct way	4	4	0	0	0	-	
xHTML-1.1-int-6	To test base Module. The base element specifies a base URL. The purpose is to check that this base URL is used in a correct way when linking to a WML page	4	4	0	0	0	-	
xHTML-1.1-int-7	To test if the interaction between WML and XHTML is handled correctly	4	4	0	0	0	-	
xHTML-1.1-int-8	To test if the interaction between WML and XHTML is handled correctly using links	4	4	0	0	0	-	
xHTML-1.1-int-9	To test List Module	4	3	1	0	0	-	

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
xHTML-1.1-int-10	To test that none terminated tags generate error messages	4	0	0	0	4	-	
xHTML-1.1-int-11	To test that tags terminated in wrong order generate error messages	4	0	0	0	4	-	

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
xHTML-1.1-con-1	To test Structure Module.	4	4	0	0	0	-	
xHTML-1.1-con-2	To test Text Module.	4	2	1	0	1	-	
xHTML-1.1-con-3	To test Hypertext Module.	4	2	1	0	1	-	
xHTML-1.1-con-4	To test List Module.	4	2	1	0	1		
xHTML-1.1-con-5	To test Basic Forms.	4	0	3	0	1		
xHTML-1.1-con-6	To test Basic Tables.	4	2	1	0	1		
xHTML-1.1-con-7	To test Image Module.	4	1	1	0	2		
xHTML-1.1-con-8	To test object Module. The Object Module is ignored, but objects with an image, which is possible to show, shall work as an image tag.	4	0	0	0	4		
xHTML-1.1-con-9	The Metainformation Module is ignored. This test checks that the element is ignored in a correct way..	4	0	0	0	4		
xHTML-1.1-con-10	To test Link Module.	4	4	0	0	0		
xHTML-1.1-con-11	To test Base Module.	4	4	0	0	0		
xHTML-1.1-con-12	To test fieldset element in Forms module.	4	3	1	0	0		
xHTML-1.1-con-13	To test optgroup element in Forms module.	4	2	2	0	0		
xHTML-1.1-con-14	To test start element on ol.	4	1	0	0	3		
xHTML-1.1-con-15	To test value attribute on li.	4	0	1	0	3		
xHTML-1.1-con-16	To test b element in Presentation module.	4	3	1	0	0		
xHTML-1.1-con-17	To test big element in Presentation module.	4	3	1	0	0		
xHTML-1.1-con-18	To test hr element in Presentation module..	4	3	1	0	0		

5.2.2.2 WCSS v1.1

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
WCSS-1.1-int-1	To test Cascade and Inheritance	4	2	1	0	1	-	
WCSS-1.1-int-2	Supported Media	4	2	1	0	1	-	
WCSS-1.1-int-3	Unsupported Media	4	4	0	0	0		
WCSS-1.1-int-4	To test Margin Shorthand	4	3	1	0	0		
WCSS-1.1-int-5	To test Padding Shorthand	4	3	1	0	0		
WCSS-1.1-int-6	To test Border Shorthand	4	3	1	0	0		
WCSS-1.1-int-7	To test Color property	4	4	0	0	0		
WCSS-1.1-int-8	To test Color property(2)	4	4	0	0	0		
WCSS-1.1-int-9	Background color	4	3	1	0	0		
WCSS-1.1-int-10	To test when the WHITE-SPACE property is set to 'pre'	3	1	2	0	0		
WCSS-1.1-int-11	To test when the TEXT-DECORATION property is set to 'underline' and/or 'blink'	4	3	1	0	0		
WCSS-1.1-int-12	To test Images as Markers	4	1	3	0	0		
WCSS-1.1-int-13	To test when VISIBILITY property is set to hidden	4	3	1	0	0		
WCSS-1.1-int-14	To test when the DISPLAY property is set to 'none'	4	3	1	0	0		
WCSS-1.1-int-15	To test when the CLEAR property is set to 'left'	4	3	1	0	0		
WCSS-1.1-int-16	To test when the VERTICAL-ALIGN property is set to 'super'.	4	1	3	0	0		
WCSS-1.1-int-17	To test style slide	4	3	1	0	0		
WCSS-1.1-int-18	To test when the -wap-marquee-dir property value is set to 'ltr'	4	3	1	0	0		
WCSS-1.1-int-19	To test margue speeds	4	3	1	0	0		
WCSS-1.1-int-20	To test when Space and Comma are separated	3	0	3	0	0		
WCSS-1.1-int-21	To test Escape characters	4	0	2	0	2		
WCSS-1.1-int-22	To test Alignment	4	2	2	0	0		
WCSS-1.1-int-23	To test shine through	4	2	2	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
WCSS-1.1-con-1	To test pattern matching.	4	0	2	0	2	-	
WCSS-1.1-con-2	To test syntax and parsing.	4	1	2	0	1	-	
WCSS-1.1-con-3	To test data types.	4	0	3	0	1		
WCSS-1.1-con-4	To test assigning property values, cascading and inheritance.	4	0	3	0	1		
WCSS-1.1-con-5	To test media types.	4	2	1	0	1		
WCSS-1.1-con-6	To test associating style sheets with XML documents.	4	2	1	0	1		
WCSS-1.1-con-7	To test pattern matching.	4	3	1	0	0		
WCSS-1.1-con-8	To test padding properties.	4	3	1	0	0		
WCSS-1.1-con-9	To test border width.	4	3	1	0	0		
WCSS-1.1-con-10	To test border colour.	4	3	1	0	0		
WCSS-1.1-con-11	To test border style.	4	0	4	0	0		
WCSS-1.1-con-12	To test border shorthand property	4	2	2	0	0		
WCSS-1.1-con-13	To test foreground colour.	4	4	0	0	0		
WCSS-1.1-con-14	To test background colour.	4	3	1	0	0		
WCSS-1.1-con-15	To test background images.	4	1	2	0	1		
WCSS-1.1-con-16	To test background shorthand property.	4	1	1	1	1		
WCSS-1.1-con-17	To test font family.	4	3	1	0	0		
WCSS-1.1-con-18	To test font style.	4	3	1	0	0		
WCSS-1.1-con-19	To test font variant.	3	1	2	0	0		
WCSS-1.1-con-20	To test font weight.	4	3	1	0	0		
WCSS-1.1-con-21	To test font size.	4	1	3	0	0		
WCSS-1.1-con-22	To test font shorthand property.	4	1	2	0	1		
WCSS-1.1-con-23	To test lists.	4	2	2	0	0		
WCSS-1.1-con-24	To test text indentation.	4	3	1	0	0		
WCSS-1.1-con-25	To test text alignment.	4	3	1	0	0		
WCSS-1.1-con-26	To test text decoration.	4	2	2	0	0		
WCSS-1.1-con-27	To test text transformation.	3	2	1	0	0		
WCSS-1.1-con-28	To test white space.	3	3	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
WCSS-1.1-con-29	To test visual effects.	3	2	1	0	0		
WCSS-1.1-con-30	To test display properties.	3	1	2	0	0		
WCSS-1.1-con-31	To test float positioning.	4	3	1	0	0		
WCSS-1.1-con-32	To test float flow control.	4	0	3	0	1		
WCSS-1.1-con-33	To test content width and height.	4	1	3	0	0		
WCSS-1.1-con-34	To test CSS Extension: Marquee.	4	1	2	0	1		
WCSS-1.1-con-35	To test CSS Extension: Access keys.	3	0	3	0	0		
WCSS-1.1-con-36	To test CSS Extension: Input.	4	0	3	0	1		

5.2.2.3   WML v1.3

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/charset/entities/1	When a named character entity is encountered (e.g. &), its associated element from the document character set replaces the character entity in the text.	4	4	0	0	0	-	
wml/charset/entities/2	When a decimal numeric character entity is encountered, its associated element from the document character set replaces the character entity in the text.	4	4	0	0	0	-	
wml/charset/entities/3	When a hexadecimal numeric character entity is encountered, its element from the document character set replaces the character entity in the text.	4	4	0	0	0		
wml/charset/model/2	When no character encoding is specified, the default character encoding can be assumed (UTF-8).	4	4	0	0	0		
wml/events/a/1	When the HREF attribute of the A element is specified, it defines the URL of a resource to be associated with the items within the A element. Selecting the contents of the A element causes the resource referenced to be loaded.	4	4	0	0	0		
wml/events/anchor/1	When an ANCHOR element is specified, its contents define a link (via a task element such as GO, PREV, or REFRESH) that is activated when the link is selected.	4	4	0	0	0		
wml/events/do/1	When a DO element is specified within a TEMPLATE element, the user agent shall behave as if the element is specified at the end of the CARD element.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/events/do/2	When a DO element is specified within a TEMPLATE element, and a DO element with the same NAME attribute is specified within a CARD element, the CARD level element overrides the TEMPLATE level element.	4	4	0	0	0		
wml/events/do/3	When a DO element has a NOOP task, it shall not be presented to the user.	4	3	1	0	0		
wml/events/do/4	When a DO element has a task other than NOOP, it shall be presented to the user.	4	3	1	0	0		
wml/events/do/5	When a DO element from a TEMPLATE is overridden by a DO element in a CARD, the DO elements from the TEMPLATE shall not be presented to the user.	4	4	0	0	0		
wml/events/do/7	When a NAME attribute of a DO element is specified, its value names the DO element. All uniquely named, active DO elements in a DECK shall be presented to the user.	4	2	2	0	0		
wml/events/do/10	User agents must accept any TYPE, but may treat any unrecognised type as the equivalent of UNKNOWN.	4	4	0	0	0		
wml/events/do/11	When the OPTIONAL attribute of a DO element is set to "false" or is omitted, the DO event must be presented to the user.	4	4	0	0	0		
wml/events/do/18	When the TYPE attribute of the DO element is "ACCEPT" the user agent must display the event as an option.	4	3	1	0	0		
wml/events/do/19	When the TYPE attribute of the DO element is "PREV" the user agent must display the event as an option.	4	4	0	0	0		
wml/events/do/20	When the TYPE attribute of the DO element is "HELP" the user agent must display the event as an option.	4	4	0	0	0		
wml/events/do/21	When the TYPE attribute of the DO element is "RESET" the user agent must display the event as an option.	4	4	0	0	0		
wml/events/do/22	When the TYPE attribute of the DO element is "DELETE" the user agent must display the event as an option.	4	4	0	0	0		
wml/events/do/23	When the TYPE attribute of the DO element is "OPTIONS" the user agent must display the event as an option.	4	3	1	0	0		
wml/events/do/24	When the TYPE attribute of the DO element is "UNKNOWN" the user agent must display the event as an option.	4	4	0	0	0		
wml/events/intrinsics/1	If the event binding element specifies an intrinsic event type which applies to its parent element, it must be ignored by the user agent.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/events/intrinsics/onevent/1	The user agent must ignore any ONEVENT element specifying a type that does not correspond to a legal intrinsic event for the immediately enclosing element.	4	4	0	0	0		
wml/events/intrinsics/onevent/onenterbackward/2	When an ONEVENT element occurs NOT immediately within a TEMPLATE or CARD element, and the TYPE attribute of the ONEVENT element is specified using the value "onenterbackward", the ONEVENT element is ignored.	4	4	0	0	0		
wml/events/intrinsics/onevent/onenterforward/1	When an ONEVENT element occurs immediately within a TEMPLATE or CARD element, and the TYPE attribute of the ONEVENT element is specified using the value "onenterforward", the ONEVENT element defines a task to be executed when the user causes the card to be entered via the GO element (or actions with identical semantics).	4	4	0	0	0		
wml/events/intrinsics/onevent/onenterforward/2	When an ONEVENT element occurs NOT immediately within a TEMPLATE or CARD element, and the TYPE attribute of the ONEVENT element is specified using the value "onenterforward", the ONEVENT element is ignored.	4	3	1	0	0		
wml/events/intrinsics/onevent/onpick/1	When an ONEVENT element occurs immediately within an OPTION element, and the TYPE attribute of the ONEVENT element is specified using the value "onpick", the ONEVENT element defines a task to be executed when the user selects or deselects the item described by the OPTION element.	4	3	1	0	0		
wml/events/intrinsics/onevent/onpick/2	When an ONEVENT element occurs NOT immediately within an OPTION element, and the TYPE attribute of the ONEVENT element is specified using the value "onpick", the ONEVENT element is ignored.	4	4	0	0	0		
wml/events/intrinsics/onevent/ontimer/1	When an ONEVENT element occurs immediately within a TEMPLATE or CARD element, and the TYPE attribute of the ONEVENT element is specified using the value "ontimer", the ONEVENT element defines a task to be executed when the TIMER expires.	4	4	0	0	0		
wml/events/intrinsics/onevent/ontimer/2	When an ONEVENT element occurs NOT immediately within a TEMPLATE or CARD element, and the TYPE attribute of the ONEVENT element is specified using the value "ontimer", the ONEVENT element is ignored.	4	3	1	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/events/postfield/1	The NAME attribute of the POSTFIELD element defines the name of a field, and the VALUE attribute defines the value to associate with that NAME. When a GO task is executed the name and value of the field are sent to the server.	4	3	0	1	0		
wml/events/setvar/2	When a SETVAR element contains a NAME attribute that evaluates to a legal variable name, the value of that variable (in the current browser context) is set to the value of the VALUE attribute.	4	2	1	1	0		
wml/events/shadowing/1	When an event-handling element is specified within a TEMPLATE element, that event-handler is available in each card of the deck.	3	2	0	1	0		
wml/events/shadowing/3	When an event-handling element is specified within a TEMPLATE element, and an event-handling element identifying the same event is specified within a CARD element, and that handler specifies its task as NOOP, there is no event handler for the event within the card. The user agent shall not provide a control for this event in this card.	4	4	0	0	0		
wml/events/shadowing/4	When a card level element binds a noop task and is not shadowed by another element the event will be masked and ignored.	4	4	0	0	0		
wml/events/tasks/go/1	When the HREF attribute of the GO element specifies a URL that names a WML card or deck, prior to displaying the card (or deck), the location of the current card is pushed onto the history stack.	3	2	0	1	0		
wml/events/tasks/go/2	When the SENDREFERER attribute of the GO element is set to true, the user agent must send the URL of the deck containing this task to the server. The URL sent must be the smallest relative URL possible if it can be relative at all.	4	3	1	0	0		
wml/events/tasks/go/3	When the METHOD attribute of the GO element is set to GET, it defines the submission method for the request.	4	4	0	0	0		
wml/events/tasks/go/4	When the METHOD attribute of the GO element is not set, the submission method for the request defaults to GET.	4	4	0	0	0		
wml/events/tasks/go/5	When the SENDREFERER attribute of the GO element is set to "false" or is not set, the user agent must not send the URL of the deck containing this task to the server.	4	4	0	0	0		
wml/events/tasks/go/8	When the METHOD attribute of the GO element is set to POST, it defines the submission method for the request.	4	3	0	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/events/tasks/go/9	When the target of a GO element of a card is contained within the current deck the user agent must ignore all postfield elements.	4	3	1	0	0		
wml/events/tasks/go/10	When the cache-control attribure is present within the GO element and its value is set to "no-cache" the client must reload the URL from the origin server.	4	3	1	0	0		
wml/events/tasks/go/12	When the HREF attribute value of a GO element is an HTTP URI, the METHOD attribute has a value of "post" and the ENCTYPE attribute is "application/x-www-form-urlencoded" the request must be performed according to these attributes.	4	3	0	1	0		
wml/events/tasks/go/14	When the ENCTYPE attributes value of a GO element is "APPLICATION/X-WWW-FORM-URLENCODED", the field names and values must be encoded using URI-escaping, and listed in the order in which the postfields are presented.	4	3	0	1	0		
wml/events/tasks/go/15	When the ENCTYPE attributes value is "APPLICATION/X-WWW-FORM-URLENCODED", the field names and values must be encoded by the name seperated from the value by "=" and name/value pairs seperated from each other by "&".	4	3	0	1	0		
wml/events/tasks/go/16	When the user agent only supports data submission as "APPLICATION/X-WWW-FORM-URLENCODED" content type, it may ignore the ENCTYPE attribute.	4	3	1	0	0		
wml/events/tasks/go/19	When the HREF attribute value of a GO element is an HTTP URI, the METHOD attribute has a value of "get" and the ENCTYPE attribute is "application/x-www-form-urlencoded" the request must be performed according to these attributes.	4	4	0	0	0		
wml/events/tasks/go/20	When the HREF attribute value of a GO element is an HTTP URI, the METHOD attribute has a value of "post" and the ENCTYPE attribute is "multipart/form-data" the request must be performed according to these attributes.	4	3	0	1	0		
wml/events/tasks/prev/1	When a PREV element is executed, the card referenced by the previous URI in the history stack is loaded (if necessary) and executed.	4	3	0	1	0		
wml/events/tasks/prev/2	When a PREV element is executed, the previous URI on the history stack is removed from the stack.	4	3	0	1	0		
wml/events/tasks/refresh/1	When a REFRESH element is executed, the current card is redisplayed.	4	3	1	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/events/tasks/refresh/2	When a REFRESH element is executed, any variables specified through enclosed SETVAR elements are set in the browser context prior to the card being redisplayed.	4	3	1	0	0		
wml/semantics/unknown/1	When a WML deck is encoded with an alternate DTD that includes elements or attributes beyond those in the WML 1.3 specification, the user agent will render the content of the unrecognized elements.	4	3	1	0	0		
wml/state/restrictions/1	Whenever a user agent navigates to a resource that was not the result of an interaction with the content in the currentcontext, the user agent must establish a fresh context for that navigation.	4	3	0	1	0		
wml/state/variables/dollarsign/1	When 2 dollar-signs in a row are encountered in a deck, they are interpreted as a single literal dollar-sign.	4	4	0	0	0		
wml/state/variables/parsing/1	All XML parsing is complete prior to variable substitution. When a dollar-sign in any form (a character entity, for example) is followed by a variable name, it is interpreted as a variable reference.	4	2	2	0	0		
wml/state/variables/setting/1	When a user commits the input in a SELECT element, the associated variable value is updated.	4	3	0	1	0		
wml/state/variables/setting/2	When a user commits the input in an INPUT element, the associated variable value is updated.	4	2	1	1	0		
wml/state/variables/substitution/1	When an undefined variable is referenced within any element or attribute with a content type of %vdata, it results in the substitution of the empty string.	4	4	0	0	0		
wml/state/variables/substitution/2	Variable names are case sensitive.	4	3	1	0	0		
wml/state/variables/substitution/3	When a variable is referenced within any element or attribute with a content type of %vdata, and that variable reference is enclosed in parenthesis using the pattern '(' name ':' conversion ')', the value of the variable is converted before it is substituted. Legal conversion values are "noesc", "escape", and "unesc" - meaning "no change", "URL-escape the value", and "URL-unescape the value" respectively.	3	1	2	0	0		
wml/state/variables/substitution/4	When a variable is referenced within any element or attribute with a content type of %vdata, and that variable reference specifies no conversion suffix, the default conversion is context sensitive. In the case of ONPICK="onenterforward", ONPICK="onenterbackward", HREF, and SRC attributes the default is escape conversion.	4	2	1	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/state/variables/substitution/6	When variables are placed in the text(#PCDATA)of a card or into %vdata and %HREF attributes in WML elements they are replaced with the variables value.	4	3	1	0	0		
wml/state/variables/substitution/7	When a string substitution occurs, the current value of the variable must not be altered.	3	2	1	0	0		
wml/state/variables/substitution/8	WML variable names consist of an US-ASCII letter or underscore followed by zero or more letters, digits or underscores. Any other characters are illegal. Any variable containing illegal characters shall not be processed by the encoder. The remainder of the card should be processed and displayed.	4	2	2	0	0		
wml/state/variables/substitution/9	When a variable is referenced within any element or attribute with a content type of %vdata, and that variable reference specifies no conversion suffix, the default is no conversion.	4	3	0	1	0		
wml/state/variables/substitution/10	When a variable is referenced within any element or attribute with a content type of %HREF, and that variable reference specifies no conversion suffix, the default is ESCAPE conversion.	4	3	0	1	0		
wml/struct/card/elements/1	When the ID attribute of the CARD element is specified as a string, it defines a name for the card. This name can then be used in the fragment portion of a URI to access the card directly.	4	4	0	0	0		
wml/struct/card/elements/2	When the NEWCONTEXT attribute of the CARD element is specified as "true", the current browser context is reinitialized when the card is entered.	4	3	0	1	0		
wml/struct/card/elements/3	When the NEWCONTEXT attribute of the CARD element is specified as "false" or is not specified, the current browser context is not initialized when the card is entered.	4	3	1	0	0		
wml/struct/card/events/1	When the ONENTERFORWARD attribute of the CARD element is specified as a URL, it defines a resource to navitage to when the CARD is entered via a GO event or the equivalent.	4	4	0	0	0		
wml/struct/card/events/2	When the ONENTERBACKWARD attribute of the CARD element is specified as a URL, it defines a resource to navigate to when the CARD is entered via a PREV event or the equivalent.	4	3	0	1	0		
wml/struct/card/events/3	When the ONTIMER attribute of the CARD element is specified as a URL, it defines a resource to navigate to when the timer for the CARD expires.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/fieldset/1	When the FIELDSET element is used the user agent shall display all input fields even if it does not support FIELDSETS.	4	4	0	0	0		
wml/struct/control/fieldset/2	When the FIELDSET element is nested, the user agent must display all of the fields.	4	4	0	0	0		
wml/struct/control/input/1	All INPUT objects that represent the input elements within the card must be initialized when the card is loaded.	4	3	1	0	0		
wml/struct/control/input/3	When the MAXLENGTH attribute of the INPUT element is specified as a number, it defines the maximum number of characters that can be entered in the input field.	4	4	0	0	0		
wml/struct/control/input/emptyok/1	When the EMPTYOK attribute of the INPUT element is specified as "true", and the FORMAT attribute of the INPUT element is also specified, the input field may be left blank by the user.	4	4	0	0	0		
wml/struct/control/input/emptyok/2	When EMPTYOK is false, input is required even if the format mask would otherwise not require it.	4	4	0	0	0		
wml/struct/control/input/emptyok/3	When EMPTYOK is true, input is not required even if the format mask would otherwise require it.	4	4	0	0	0		
wml/struct/control/input/format/1	When the FORMAT attribute of the INPUT element is specified as "*A", the text input is limited to upper-case letter, symbol or punctuation characters. Numeric characters are excluded	4	4	0	0	0		
wml/struct/control/input/format/2	When the FORMAT attribute of the INPUT element is specified as "*a", the text input is limited to lower-case letter, symbol or punctuation characters.	4	4	0	0	0		
wml/struct/control/input/format/3	When the FORMAT attribute of the INPUT element is specified as "*N", the text input is limited to numeric characters.	4	4	0	0	0		
wml/struct/control/input/format/4	When the FORMAT attribute of the INPUT element is specified as "*X", the text input is limited to entry of any uppercase letter, numeric character, symbol, or punctuation character.	4	3	1	0	0		
wml/struct/control/input/format/5	When the FORMAT attribute of the INPUT element is specified as "*x", the text input is limited to entry of any lowercase letter, numeric character, symbol, or punctuation character.	4	2	2	0	0		
wml/struct/control/input/format/7	When the FORMAT attribute of the INPUT element is specified as a backslash "\" followed by a character c, that character is displayed in the input field at the position indicated.	4	2	2	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/input/format/8	When the FORMAT attribute 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.	4	2	2	0	0		
wml/struct/control/input/format/9	When the FORMAT attribute of the INPUT element is specified as "*M", the text input is limited to any characters, but the user-agent may assume that the input is in upper-case.	4	3	1	0	0		
wml/struct/control/input/format/10	When the FORMAT attribute of the INPUT element is specified as "*m", the text input is limited to any characters, but the user-agent may assume that the input is in lower-case.	4	2	2	0	0		
wml/struct/control/input/format/11	When the FORMAT attribute of the INPUT element is specified as "nM", where n is a digit from 1-9 and f is one of the format characters "aAmMNXx", the number of input characters is limited to n and the type is limited to f.	4	3	1	0	0		
wml/struct/control/input/format/12	When the FORMAT attribute of the INPUT element is specified as an invalid mask it must be ignored by the user agent.	4	3	1	0	0		
wml/struct/control/input/name/1	The NAME attribute of the INPUT element specifies the name of a variable in which the input value is stored. If this variable is set when the INPUT element is displayed, its value is used to initialize the content of the input field.	4	3	1	0	0		
wml/struct/control/input/type/1	When the TYPE attribute of the INPUT element is specified as "text", it indicates that the input field is a text entry box, and that the input is to be displayed to the user in readable form.	4	4	0	0	0		
wml/struct/control/input/type/2	When the TYPE attribute of the INPUT element is specified as "password", it indicates that the input field is a password entry box, and that the input is to be displayed to the user in an illegible form.	4	3	1	0	0		
wml/struct/control/input/type/3	When the TYPE attribute of the INPUT element is not specified, it defaults to "text". This indicates that the input field is a text entry box, and that the input is to be displayed to the user in readable form.	4	4	0	0	0		
wml/struct/control/input/value/1	When the VALUE attribute of the INPUT element is specified, it defines the default value for the variable named by the NAME attribute. When the INPUT element is displayed and the variable named by the NAME attribute is not set, that variable is initialized with the value of the VALUE attribute.	4	3	1	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/input/value/2	When the VALUE attribute of the INPUT element is specified, it defines the default value for the variable named by the NAME attribute. When the INPUT element is displayed and the variable named by the NAME attribute is already set, the value of the VALUE attribute is ignored.	4	4	0	0	0		
wml/struct/control/select/element/iname/1	When the INAME attribute of the SELECT element is specified, it defines the name of a variable in which the index of the user selected option(s) will be stored.	4	3	0	1	0		
wml/struct/control/select/element/iname/2	When the IVALUE attribute of the SELECT element is set, it defines a default index value with which to select options within the SELECT element. If the INAME attribute is also specified, and the variable named by INAME already contains a value, the IVALUE attribute value is ignored.	4	3	1	0	0		
wml/struct/control/select/element/iname/3	When the INAME attribute of the SELECT element is set to a string, it defines the name of a variable in which to record the user's selection(s). When the MULTIPLE attribute of this SELECT element is set to TRUE, and multiple options are selected, the value of the variable named by INAME is a semi-colon separated list of selected option indices.	4	3	1	0	0		
wml/struct/control/select/element/iname/4	When the INAME attribute of the SELECT element is set to a string, it defines the name of a variable in which to record the user's selection(s). If the variable named by the INAME attribute is set when the element is displayed, the value of the variable is used as indices to pre-select option(s) within the SELECT element.	4	3	1	0	0		
wml/struct/control/select/element/iname/5	When the INAME attribute of the SELECT element is specified with a variable name, and the value of that variable is 0, it indicates that no options within the SELECT element are selected.	4	2	1	1	0		
wml/struct/control/select/element/iname/6	When the INAME attribute of the SELECT element is specified with a variable name, and that variable is not set, and the VALUE or IVALUE attribute is not specified, and the MULTIPLE attribute is set to "TRUE", the selected index defaults to 0.	4	2	1	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/select/element/iname/7	When the INAME attribute of the SELECT element is specified with a variable name, and that variable is set to a value greater than the largest index of the enclosed OPTION elements, and the MULTIPLE attribute is set to "TRUE", no option is selected.	4	2	1	1	0		
wml/struct/control/select/element/iname/8	When the INAME attribute of the SELECT element is not specified, and the NAME attribute is specified as a variable name, and that variable is set, the value of that variable is used to select option(s).	4	2	1	1	0		
wml/struct/control/select/element/iname/9	When the INAME attribute of the SELECT element is not specified, and the NAME attribute is specified as a variable name, and that variable is set, but the value to which it is set does not match the VALUE attribute of an enclosed OPTION element, and the MULTIPLE attribute is set to "TRUE", no option is selected.	4	2	1	1	0		
wml/struct/control/select/element/iname/10	When the INAME attribute of the SELECT element is not specified, and the NAME attribute is specified as a variable name, and that variable is not set, and the MULTIPLE attribute is set to FALSE, the first option is selected.	4	3	0	1	0		
wml/struct/control/select/element/iname/11	When the INAME attribute of the SELECT element is not specified, and the NAME attribute is specified as a variable name, and that variable is set, but the value to which it is set does not match the VALUE attribute of an enclosed OPTION element, and the MULTIPLE attribute is set to "FALSE", the first option is selected.	4	3	0	1	0		
wml/struct/control/select/element/iname/12	When the INAME attribute of the SELECT element is specified with a variable name, and that variable is set to a value greater than the largest index of the enclosed OPTION elements, and the MULTIPLE attribute is set to "FALSE", the first option is selected.	4	2	1	1	0		
wml/struct/control/select/element/iname/13	When the NAME and INAME variables are updated the user agent must not display any side effects.	4	2	1	1	0		
wml/struct/control/select/element/ivalue/1	When the IVALUE attribute of the SELECT element is set, it defines a default value with which to select options within the SELECT element. If the MULTIPLE attribute of the SELECT element is set to TRUE, this default value is interpreted as a semi-colon separated list of option indices to select.	4	2	1	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/select/element/multiple/1	When the MULTIPLE attribute of the SELECT element is set to TRUE, the user is permitted to select multiple options from within the element.	4	2	1	1	0		
wml/struct/control/select/element/multiple/2	When the MULTIPLE attribute of the SELECT element is set to FALSE, the user is permitted to select only a single option from within the element.	4	2	1	1	0		
wml/struct/control/select/element/multiple/3	When the MULTIPLE attribute of the SELECT element is not specified, the user is permitted to select only a single option from within the element.	4	2	1	1	0		
wml/struct/control/select/element/name/1	When the NAME attribute of the SELECT element is set to a legal variable name, it defines the name of a variable in which to record the user's selection(s).	3	2	0	1	0		
wml/struct/control/select/element/name/2	When the NAME attribute of the SELECT element is set to a string, it defines the name of a variable in which to record the user's selection(s). When the MULTIPLE attribute of this SELECT element is set to TRUE, and multiple options are selected, the value of the variable named by NAME is a semi-colon separated list of values.	4	2	1	1	0		
wml/struct/control/select/element/name/3	When the NAME attribute of the SELECT element is set to a string, it defines the name of a variable in which to record the user's selection(s). If the variable named by the NAME attribute is set when the element is displayed, the value of the variable is used to pre-select option(s) within the SELECT element.	4	2	1	1	0		
wml/struct/control/select/element/value/1	When the VALUE attribute of the SELECT element is set, it defines a default value with which to select options within the SELECT element.	4	2	1	1	0		
wml/struct/control/select/element/value/2	When the VALUE attribute of the SELECT element is set, it defines a default value with which to select options within the SELECT element. If the NAME attribute is also specified, and the variable named by NAME already contains a value, the VALUE attribute value is ignored.	4	3	0	1	0		
wml/struct/control/select/element/value/3	When the VALUE attribute of the SELECT element is set, it defines a default value with which to select options within the SELECT element. If the MULTIPLE attribute of the SELECT element is set to TRUE, this default value is interpreted as a semi-colon separated list of values to select.	4	2	1	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/control/select/optgroup/1	When the OPTGROUP element is encountered, the implementation must allow the elements to be grouped and displayed in a hierarchical manor.	4	2	1	1	0		
wml/struct/control/select/optgroup/2	When a user agent encounters an OPTGROUP element it must display the elements children in readable manner even if OPTGROUP is not supported.	4	2	1	1	0		
wml/struct/control/select/option/1	When the VALUE attribute of the OPTION element is set to a string, it defines the value to be used when setting the variable named by the NAME attribute of the enclosing SELECT element.	4	2	0	2	0		
wml/struct/control/select/option/2	When the ONPICK attribute of the OPTION element is set to a URL, and the MULTIPLE attribute of the enclosing SELECT element is set to TRUE, it defines a task that is to be executed whenever the option is selected or deselected.	4	2	0	2	0		
wml/struct/control/select/option/3	When the ONPICK attribute of the OPTION element is set to a URL, and the MULTIPLE attribute of the enclosing SELECT element is set to FALSE or not specified, it defines a task that is to be executed whenever the option is selected.	4	2	1	1	0		
wml/struct/control/select/option/4	When the OPTION element contains PCDATA, that data is presented as the description of the option.	4	2	1	1	0		
wml/struct/control/select/option/5	When the OPTION element contains a value with a variable reference, the variable is evaluated before the name variable is set.	4	2	1	1	0		
wml/struct/control/tabindex/1	When the TABINDEX attribute of an element is specified as a number, it defines the relative position in the "tabbing order" within the card - where numerically greater numbers indicate a later position than numerically lesser numbers.	4	3	0	1	0		
wml/struct/control/tabindex/2	When the TABINDEX attribute of an input element is not specified, the user agent assigns it a position in the "tabbing order" that is later than all user-specified TABINDEX attributes within the card.	4	3	0	1	0		
wml/struct/head/access/1	When an ACCESS element is in the HEAD element of a deck and that deck is entered, the user agent checks to ensure that the new deck can be accessed from the current deck.	4	2	1	1	0		
wml/struct/head/access/2	When the DOMAIN attribute of the ACCESS element (of a new deck) is not set, it defaults to the domain of the current deck.	4	2	1	1	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/head/access/3	When the PATH attribute of the ACCESS element (of a new deck) is not set, it defaults to the value "/".	4	3	0	1	0		
wml/struct/head/access/4	When the DOMAIN attribute of the ACCESS element (of a new deck) is set to a period separated domain suffix, the domain of the current deck is compared to this suffix - where each element specified must match exactly for access to be granted.	4	3	0	1	0		
wml/struct/head/access/5	When the PATH attribute of the ACCESS element (of a new deck) is specified as an absolute path, it defines the path prefix that is permitted access to the deck - where each element of the prefix must match that of the current deck exactly for access to be granted.	4	3	1	0	0		
wml/struct/head/access/6	When a deck contains more than one ACCESS element the user agent shall display an error, along with a means to continue browser navigation.	4	3	1	0	0		
wml/struct/head/access/7	When a user agent encounters a relative path in an ACCESS element it shall convert it to an absolute path. Then it should perform prefix matching against the PATH attribute	4	3	1	0	0		
wml/struct/images/1	When the ALT attribute of the IMG element is specified as a string, it defines alternate text that is displayed when the image cannot be loaded.	4	4	0	0	0		
wml/struct/images/2	When the SRC attribute of the IMG element is specified as a URL, it defines the location of the image data.	4	4	0	0	0		
wml/struct/images/3	When the VSPACE attribute of the IMG element is specified as a length, it defines the vertical whitespace to be inserted above and below the image.	4	2	2	0	0		
wml/struct/images/4	When the HSPACE attribute of the IMG element is specified as a length, it defines the horizontal whitespace to be inserted to the left and right of the image.	4	2	2	0	0		
wml/struct/images/5	When the VSPACE attribute of the IMG element is not specified, it defaults to zero length.	4	4	0	0	0		
wml/struct/images/6	When the HSPACE attribute of the IMG element is not specified, it defaults to zero length.	4	4	0	0	0		
wml/struct/images/7	When the ALIGN attribute of the IMG element is specified as TOP, it means that the top of the image is aligned vertically with the top of the current text line.	4	3	1	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/images/8	When the ALIGN attribute of the IMG element is specified as BOTTOM, it means that the bottom of the image is aligned vertically with the current baseline.	4	3	1	0	0		
wml/struct/images/9	When the ALIGN attribute of the IMG element is specified as MIDDLE, it means that the vertical center of the image is aligned vertically with the center of the current text line.	4	3	1	0	0		
wml/struct/images/11	When the ALT attribute of the IMG element is specified as a string, it defines alternate text that is displayed when the user agent does not support the image format.	4	4	0	0	0		
wml/struct/template/1	When an event binding (e.g. a DO or ONEVENT element) is specified within a TEMPLATE element, that specification applies to all cards in the deck.	4	4	0	0	0		
wml/struct/template/2	When a DO element is specified within a TEMPLATE element, and a DO element with the same NAME attribute value is specified within a CARD element, the DO element within the CARD element takes precedence.	4	3	1	0	0		
wml/struct/text/br/1	When a BR element is encountered, the user agent breaks the current line. Subsequent text in the paragraph is continued on the following line.	4	4	0	0	0		
wml/struct/text/emphasis/b/1	When the B inline element is used, the text enclosed within the B element is displayed in a bold typeface.	4	3	1	0	0		
wml/struct/text/emphasis/big/1	When the BIG inline element is used, the text enclosed within the BIG element is displayed in a larger typeface.	4	3	1	0	0		
wml/struct/text/emphasis/em/1	When the EM inline element is used, the text enclosed within the EM element is displayed in an alternative typeface to give emphasis to the text.	4	3	1	0	0		
wml/struct/text/emphasis/i/1	When the I inline element is used, the text enclosed within the I element is displayed in an italicized typeface.	4	3	1	0	0		
wml/struct/text/emphasis/small/1	When the SMALL inline element is used, the text enclosed within the SMALL element is displayed in a smaller typeface.	4	3	1	0	0		
wml/struct/text/emphasis/strong/1	When the STRONG inline element is used, the text enclosed within the STRONG element is displayed in an alternative typeface to give strong emphasis to the text	4	3	1	0	0		
wml/struct/text/emphasis/u/1	When the U inline element is used, the text enclosed within the U element is displayed underlined.	4	3	1	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/text/p/1	When a non-breaking space entity is encountered, the user agent does not treat it as an inter-word space.	4	2	2	0	0		
wml/struct/text/p/5	When a P element is empty or contains only insignificant white space, it must not impact line wrap mode.	4	4	0	0	0		
wml/struct/text/p/align/1	When the ALIGN attribute of the P element is specified as "left", the text of the line is aligned on the left side of the display.	4	4	0	0	0		
wml/struct/text/p/align/2	When the ALIGN attribute of the P element is specified as "center", the text of the line is centered on the display.	4	3	1	0	0		
wml/struct/text/p/align/3	When the ALIGN attribute of the P element is specified as "right", the text of the line is aligned on the right side of the display.	4	3	1	0	0		
wml/struct/text/p/align/4	When the ALIGN attribute of the P element is not specified, text is aligned with the left side of the display.	4	4	0	0	0		
wml/struct/text/p/mode/1	When the MODE attribute of the P element is specified as "wrap", the text of the line is broken onto multiple display lines at whitespace characters as needed.	4	3	1	0	0		
wml/struct/text/p/mode/2	When the MODE attribute of the P element is specified as "nowrap", the text of the line is displayed on a single line.	4	3	1	0	0		
wml/struct/text/p/mode/3	When the MODE attribute of the P element is not specified, it defaults to the MODE of the previous paragraph.	4	3	1	0	0		
wml/struct/text/p/mode/4	When the MODE attribute of the P element is not specified, and the P element is the first P element of the CARD, it defaults to "wrap" mode.	4	3	1	0	0		
wml/struct/text/table/1	When the COLUMNS attribute of the TABLE element is specified as a non-negative integer, it defines the number of columns that each row of the table shall contain. Rows with fewer than the specified number of columns have empty columns added to their ends.	4	4	0	0	0		
wml/struct/text/table/2	When the COLUMNS attribute of the TABLE element is specified as a non-negative integer, it defines the number of columns that each row of the table shall contain. Rows with more than the specified number of columns have any additional data collected into the last defined column of the row, with each column's data separated from the others by a single inter-word space.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/text/table/3	When the ALIGN attribute of the TABLE element is specified as a sequence of the letters "R", "L", or "C", one for each column in the table, it defines the default alignment for data in each column as right, left, or center, respectively.	4	3	1	0	0		
wml/struct/text/table/4	When the ALIGN attribute of the TABLE element is specified as a sequence of the letters "R", "L", or "C", but there are fewer letters than columns, the contents of the unspecified columns get the default alignment (left for left-to-right languages, right for right-to-left languages).	4	3	1	0	0		
wml/struct/text/table/5	When the ALIGN attribute of the TABLE element is not specified, the contents of the columns get the default alignment (left for left-to-right languages, right for right-to-left languages).	4	4	0	0	0		
wml/struct/text/table/7	When the ALIGN attribute of the TABLE element is specified as "D" the contents of that column will be default aligned (left for left-to-right languages, right for right-to-left languages). All extra designators should be ignored.	4	4	0	0	0		
wml/struct/text/table/9	When the ALIGN attribute of the TABLE element is specified as an unknown designator the contents of that column will be default aligned (left for left-to-right languages, right for right-to-left languages).	4	4	0	0	0		
wml/struct/text/td/1	When the TD element is specified, it defines the contents of a single cell of a row of a table. The contents of the element are rendered in the cell.	4	4	0	0	0		
wml/struct/text/td/2	When the TD element is specified and but has no contents, the element must still be counted as a cell for rendering purposes.	4	4	0	0	0		
wml/struct/text/tr/1	When a table row is empty it is significant and must not be ignored.	4	2	2	0	0		
wml/struct/text/whitespace/2	When the user agent encounters white space before or after an element or attribute it should be ignored.	4	3	1	0	0		
wml/struct/timer/1	When the VALUE attribute of the TIMER element is specified as a positive integer value and the NAME value is not specified, the VALUE attribute defines the default starting value for the timer. Upon entering the card, the timer is started with this value. If the card is not exited prior to the expiration of the timer, the ONTIMER intrinsic event is delivered.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/struct/timer/2	When the NAME attribute of the TIMER element is specified as a string, it defines a variable in which the timer value is to be stored.	4	3	1	0	0		
wml/struct/timer/3	When the NAME attribute of the TIMER element is specified as a string, it defines a variable name. If that variable is set, the timer is initialized with the value of the variable.	4	3	1	0	0		
wml/struct/timer/4	When the NAME attribute of the TIMER element is specified as a string, it defines a variable name. If that variable is not set and the VALUE attribute of the TIMER element is specified, the variable is set to the value of the VALUE attribute, and then the variable is used to initialize the timer.	4	3	1	0	0		
wml/struct/timer/5	When the NAME attribute of the TIMER element is specified as a string, it defines a variable name. If that variable is set to zero, the timer is disabled.	4	4	0	0	0		
wml/struct/timer/7	When the value of the timeout is not a positive integer, the user agent must ignore the TIMER element.	4	3	1	0	0		
wml/syntax/attributes/1	Element attribute values must be enclosed in quotation marks. When an attribute value is enclosed in single quotation marks ('), the value may contain double-quotation marks (").	4	2	2	0	0		
wml/syntax/attributes/2	Element attribute values must be enclosed in quotation marks. When an attribute value is enclosed in double quotation marks ("), the value may contain single quotation marks (').	4	4	0	0	0		
wml/syntax/attributes/3	Element attribute values may contain character entities. Entity transformation into the document character set shall not effect the interpretation of quotation marks enclosing attribute values, nor shall it effect the parsing of attribute values.	4	4	0	0	0		
wml/syntax/cdata/1	When data is enclosed within a CDATA block (<![CDATA[data]] >), the data is handled as literal text and presented to the user "as is".	4	2	2	0	0		
wml/syntax/comments/1	When an XML comment (<!-- comment -->) is included in a deck, the contents of that comment shall not be presented to the user.	4	3	1	0	0		
wml/syntax/elements/1	When an element is declared EMPTY in the WML DTD, it may be specified using the shorthand notation <ELEMENT/>. User agents shall treat documents that use this convention and are otherwise conforming as well formed.	4	4	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
wml/syntax/elements/2	When an element is not declared EMPTY in the WML DTD, and such an element has no contents, it may be specified as <ELEMENT/> or as <ELEMENT></ELEMENT>. User agents shall treat documents that use either convention and are otherwise conforming as well formed.	4	3	1	0	0		
wml/syntax/entities/1	When a numeric character entity is enountered, it is translated into the corresponding character in the document character set.	4	3	1	0	0		
wml/syntax/entities/2	When a named character entity is enountered, it is translated into the corresponding character in the document character set.	4	4	0	0	0		
wml/syntax/pi/1	When a processing instruction (<? instruction ?>) is encountered, it is ignored unless it is defined in the XML 1.0 Recommendation.	4	3	1	0	0		
wml/syntax/variables/1	When the symbol "$", followed by an identifier, is encountered in a %PCDATA or %vdata context, it is a variable reference. The variable reference is replaced with the value of the variable thus named.	4	2	2	0	0		
wml/syntax/variables/2	When the symbol "$", followed by an identifier enclosed in parenthesis, is encountered in a %PCDATA or %vdata context, it is a variable reference. The variable reference is replaced with the value of the variable thus named.	4	3	1	0	0		
wml/syntax/variables/3	When the symbol "$", followed by an identifier, a colon ":" and a conversion specifier, enclosed in parenthesis, is encountered in a %PCDATA or %vdata context, it is a variable reference and an indicator as to how the variable value should be converted. The variable reference is replaced with the value of the variable thus named, converted as specified.	4	2	2	0	0		
wml/syntax/variables/4	When the symbol "$$" is encountered in a %PCDATA or %vdata context, it is an escaped dollar-sign. The escaped dollar-sign is replaced with a dollar-sign.	4	4	0	0	0		
wml/urls/fragment/1	When a URL does not include a fragment identifier (a hash mark followed by a name), the URL references a deck.	4	4	0	0	0		
wml/urls/fragment/2	When a URL contains a fragment identifier (a hash mark followed by a name), the URL references an individual card within a WML deck.	4	4	0	0	0		
wml/urls/relative/1	When a relative URL is encountered, it is resolved against the Base URL of the deck.	4	4	0	0	0		

5.2.2.4   ESMP v1.0

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
ESMP-1.0-int-1	To test the support for Inline Script Execution	0	0	0	0	0	-	
ESMP-1.0-int-2	To test the support for Deferred Script Execution	0	0	0	0	0	-	
ESMP-1.0-int-3	To test font the support for File Based Script Execution	0	0	0	0	0		
ESMP-1.0-int-4	To test the semicolon support at end of statements	0	0	0	0	0		
ESMP-1.0-int-5	To test the support for EvalError exceptio	0	0	0	0	0		
ESMP-1.0-int-6	To test the Native Object Support – String Object	0	0	0	0	0		
ESMP-1.0-int-7	To test the Native Object Support – Error Object	0	0	0	0	0		
ESMP-1.0-int-8	To test the support for DOM2 compliant event binding	0	0	0	0	0		
ESMP-1.0-int-9	To test the Host Object Support – parent global Object	0	0	0	0	0		
ESMP-1.0-int-10	To test the Host Object Support – navigator Object	0	0	0	0	0		
ESMP-1.0-int-11	To test the Host Object Support – history Object.	0	0	0	0	0		
ESMP-1.0-int-12	To test the Host Object Support – location Object	0	0	0	0	0		
ESMP-1.0-int-13	To test the UTF-16 Code point support	0	0	0	0	0		
ESMP-1.0-int-14	To test the UTF-8 Code point support	0	0	0	0	0		
ESMP-1.0-int-15	To test the semicolon support at end of statements	0	0	0	0	0		
ESMP-1.0-int-16	To test the ECMAScript Language Syntax and semantics	0	0	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
ESMP-1.0-con-1	To test the ECMAscript Type Support.	0	0	0	0	0	-	
ESMP-1.0-con-2	To test the UTF-16 Code point support.	0	0	0	0	0	-	
ESMP-1.0-con-3	To test the UTF-8 Code point support.	0	0	0	0	0		
ESMP-1.0-con-4	To test the character indexing support.	0	0	0	0	0		
ESMP-1.0-con-5	To test the semicolon support at end of statements.	0	0	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
ESMP-1.0-con-6	To test the IEEE 754 64 Bit Float Support with accuracy to at least 14 digits	0	0	0	0	0		
ESMP-1.0-con-7	To test the ECMAScript Language Syntax and semantics.	0	0	0	0	0		
ESMP-1.0-con-8	To test the support for eval().	0	0	0	0	0		
ESMP-1.0-con-9	To test the support for EvalError exception.	0	0	0	0	0		
ESMP-1.0-con-10	To test the support for dynamic function creation.	0	0	0	0	0		
ESMP-1.0-con-11	To test the support for the with statement.	0	0	0	0	0		
ESMP-1.0-con-12	To test the support for dynamic modification of built-in objects.	0	0	0	0	0		
ESMP-1.0-con-13	To test the version properties for all native objects	0	0	0	0	0		
ESMP-1.0-con-14	To test that all non-native, built-in objects are enumerable.	0	0	0	0	0		
ESMP-1.0-con-15	To test that All native, built-in objects are enumerable.	0	0	0	0	0		
ESMP-1.0-con-16	To test the Native Object Support – Global Object.	0	0	0	0	0		
ESMP-1.0-con-17	To test the Native Object Support – Array Object.	0	0	0	0	0		
ESMP-1.0-con-18	To test the Native Object Support – String Object.	0	0	0	0	0		
ESMP-1.0-con-19	To test the Native Object Support – Regular Expression Object.	0	0	0	0	0		
ESMP-1.0-con-20	To test the Native Object Support – Boolean Object.	0	0	0	0	0		
ESMP-1.0-con-21	To test the Native Object Support – Number Object.	0	0	0	0	0		
ESMP-1.0-con-22	To test the Native Object Support – Math Object	0	0	0	0	0		
ESMP-1.0-con-23	To test the Native Object Support – Date Object.	0	0	0	0	0		
ESMP-1.0-con-24	To test the Native Object Support – Error Object.	0	0	0	0	0		
ESMP-1.0-con-25	To test the Native Object Support – Object Creation.	0	0	0	0	0		
ESMP-1.0-con-26	To test the Native Object Support – Function Object - dynamic function construction.	0	0	0	0	0		
ESMP-1.0-con-27	To test the support for Native Error Types.	0	0	0	0	0		
ESMP-1.0-con-28	To test the support for Inline Script Execution.	0	0	0	0	0		
ESMP-1.0-con-29	To test the support for Deferred Script Execution.	0	0	0	0	0		
ESMP-1.0-con-30	To test font the support for File Based Script Execution.	0	0	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
ESMP-1.0-con-31	To test the abnormal termination error reporting.	0	0	0	0	0		
ESMP-1.0-con-32	To test the support for Aborted Script completion.	0	0	0	0	0		
ESMP-1.0-con-33	To test the support for XHTML Events.	0	0	0	0	0		
ESMP-1.0-con-34	To test the support for DOM2 compliant event binding.	0	0	0	0	0		
ESMP-1.0-con-35	To test the Host Object Support – parent global Object.	0	0	0	0	0		
ESMP-1.0-con-36	To test the Host Object Support – navigator Object.	0	0	0	0	0		
ESMP-1.0-con-37	To test the Host Object Support – history Object.	0	0	0	0	0		
ESMP-1.0-con-38	To test the Host Object Support – location Object.	0	0	0	0	0		
ESMP-1.0-con-39	To test the Host Object Support – Basic document Object.	0	0	0	0	0		
ESMP-1.0-con-40	To test the XML DOM Support.	0	0	0	0	0		
ESMP-1.0-con-41	To test the DOM Object Support – Exception Object.	0	0	0	0	0		
ESMP-1.0-con-42	To test the DOM Object Support – Node Object – Property Support.	0	0	0	0	0		
ESMP-1.0-con-43	To test the DOM Object Support – Node Object – Data Interrogation Methods.	0	0	0	0	0		
ESMP-1.0-con-44	To test the DOM Object Support – Structural Mutation.	0	0	0	0	0		
ESMP-1.0-con-45	To test the DOM Object Support – Node Object – Structural Modification Mutation Methods.	0	0	0	0	0		
ESMP-1.0-con-46	To test the DOM Object Support – document Object - Data Interrogation Methods.	0	0	0	0	0		
ESMP-1.0-con-47	To test the DOM Object Support – document Object – Structural Modification Mutation Methods.	0	0	0	0	0		
ESMP-1.0-con-48	To test the DOM Object Support – NodeLlist Object.	0	0	0	0	0		
ESMP-1.0-con-49	To test the DOM Object Support – Element Object – Property Support.	0	0	0	0	0		
ESMP-1.0-con-50	To test the DOM Object Support – Element Object – Data Interrogation Methods.	0	0	0	0	0		
ESMP-1.0-con-51	To test the DOM Object Support – Element Object – Data Modification Methods.	0	0	0	0	0		
ESMP-1.0-con-52	To test the DOM Object Support – Element Object – Structural Modification Mutation Methods.	0	0	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
ESMP-1.0-con-53	To test the DOM Object Support – Text Object.	0	0	0	0	0		
ESMP-1.0-con-54	To test the XHTML DOM Support.	0	0	0	0	0		
ESMP-1.0-con-55	To test the XHTML DOM Support – XHTML Document Object.	0	0	0	0	0		
ESMP-1.0-con-56	To test the XHTML DOM Support – Link Object.	0	0	0	0	0		
ESMP-1.0-con-57	To test the XHTML DOM Support – Image Object.	0	0	0	0	0		
ESMP-1.0-con-58	To test the XHTML DOM Support – Form Object.	0	0	0	0	0		
ESMP-1.0-con-59	To test the XHTML DOM Support – Text Input Object.	0	0	0	0	0		
ESMP-1.0-con-60	To test the XHTML DOM Support – Textarea Input Object.	0	0	0	0	0		
ESMP-1.0-con-61	To test the XHTML DOM Support – Password Input Object.	0	0	0	0	0		
ESMP-1.0-con-62	To test the XHTML DOM Support – Radio Input Object.	0	0	0	0	0		
ESMP-1.0-con-63	To test the XHTML DOM Support – Checkbox Input Object.	0	0	0	0	0		
ESMP-1.0-con-64	To test the XHTML DOM Support – Submit Object.	0	0	0	0	0		
ESMP-1.0-con-65	To test the XHTML DOM Support – Reset Object.	0	0	0	0	0		
ESMP-1.0-con-66	To test the XHTML DOM Support – Select Element Object.	0	0	0	0	0		
ESMP-1.0-con-67	To test the XHTML DOM Support – Option Element Object.	0	0	0	0	0		
ESMP-1.0-con-68	To test the XHTML DOM Support – Screen Object.	0	0	0	0	0		

5.2.2.5 UAProf 2.0

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
UAPROF-2.0-int-01	The purpose of this test is to check that the User Agent Profile transport is correct. The profile MUST be transported using WSP. Upon opening a WSP session with a WAP gateway, the UAProf-aware client conveys its profile information within the WSP Connect request	0	0	0	0	0	-	
UAPROF-2.0-int-02	The purpose of this test case is to check that the User Agent Profile transport is correct. The profile MUST be transported using W-HTTP. In the case where the mobile terminal supports wireless profiled HTTP (W-HTTP) it may transmit the CPI data in the x-wap-profile header with each HTTP request that is made	0	0	0	0	0	-	
UAPROF-2.0-int-03	The purpose of this test case is to check that User Agent Profile data is translated correctly by the WAP gateway/proxy. From the mobile client to the WAP gateway/proxy, the User Agent profile data is transferred over WSP. From the gateway to the origin server, the User Agent Profile is transported over the Internet. In the WAP architecture specification , and as specified in Section 8.2.3.3, HTTP is assumed to be the transport mechanism for Internet-related information, using the HTTP 1.1 mechanisms for header management and caching.	0	0	0	0	0		
UAPROF-2.0-int-04	The purpose of this test case is to check that User Agent Profile data is forwarded correctly by the WAP gateway/proxy. From the mobile client to the WAP gateway/proxy, the User Agent profile data is transferred over W-HTTP. From the gateway to the origin server, the User Agent Profile is transported over the Internet. In the WAP architecture specification , and as specified in Section 8.2.3.3, HTTP is assumed to be the transport mechanism for Internet-related information, using the HTTP 1.1 mechanisms for header management and caching .	0	0	0	0	0		

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
UAPROF-2.0-int-05	The purpose of this test is to check that the transport of a User Agent Profile-Diff header is correct. The profile change is transported using WAP/WSP	0	0	0	0	0		
UAPROF-2.0-int-06	The purpose of this test is to check that the transport of an x-wap-profile-diff is correct. The profile change is transported using W-HTTP.	0	0	0	0	0		
UAPROF-2.0-int-07	The purpose of this test case is to check that Profile-Diff is translated correctly by the WAP gateway/proxy. From the mobile client to the WAP gateway/proxy, the data is transferred over WSP. From the gateway to the origin server, the data is transported over the Internet. In the WAP architecture specification , and as specified in HTTP is assumed to be the transport mechanism for Internet-related information, using the HTTP 1.1 mechanisms for header management and caching.	0	0	0	0	0		
UAPROF-2.0-int-08	The purpose of this test case is to check that x-wap-profile-diff data is forwarded correctly by the WAP gateway/proxy. From the mobile client to the WAP gateway/proxy, the data is transferred over W-HTTP. From the gateway to the origin server, the data is transported over the Internet. In the WAP architecture specification, and as specified in HTTP is assumed to be the transport mechanism for Internet-related information, using the HTTP 1.1 mechanisms for header management and caching.	0	0	0	0	0		

Note: the following tests are incorrectly labelled in the OMA-ETS-UAPROF-V2_0-20041118-A.doc as **xHTML-1.1-con-1, xHTML-1.1-con-2** & **xHTML-1.1-con-1.**

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
UAPROF-2.0-con-01	The purpose of this test is to validate the UAProf and CC/PP format constraints of the User Agent Profile sent by the Client device.	0	0	0	0	0	-	
UAPROF-2.0-con-02	The purpose of this test is to validate the RDF format of the User Agent Profile sent by the Client device.	0	0	0	0	0	-	
UAPROF-2.0-con-03	The purpose of this test is to validate the datatype definitions of the User Agent Profile sent by the Client device.	0	0	0	0	0		

5.2.3   Problem Reports

During the activities for TestFest9, the following problem reports were generated relating to the test materials and test process:

PR Number	Affecting	Description	Test Case reference / Specification reference
TBC	ETS	In [xHTML_Suite_1], test case 45.07.47 contains an invalid assertion	xHTML-1.1-con-5 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test case 45.09.04 contains an invalid assertion	xHTML-1.1-con-7 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test case 45.13.09 contains an invalid assertion	xHTML-1.1-con-15 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test case 45.13.10 contains an invalid assertion	xHTML-1.1-con-14 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test cases 45.09.09 and 45.09.11 contains invalid assertions	xHTML-1.1-con-7 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test case 45.09.18 contain an invalid assertions	xHTML-1.1-con-7 in ETS for XHTML 1.1
TBC	ETS	In [xHTML_Suite_1], test case 45.07.09 contain an invalid assertion	xHTML-1.1-con-5 in ETS for XHTML 1.1
TBC	Test Content	In [xHTML_Suite_1], the test file used in 45.07.42 is invalid	xHTML-1.1-con-5 in ETS for XHTML 1.1
TBC	ETS	The test procedure for xHTML-1.1-con-2 contains references to incorrect test suite numbers.	xHTML-1.1-con-2 in ETS for XHTML 1.1
TBC	ETS	The test procedure for WCSS-1.1-int-2 contains references to incorrect test suite number.	WCSS-1.1-int-2 in ETS for WCSS 1.1
TBC	Test Content	Test files [WCSS_Suite] 45.15.06 and 45.28.13 contain the same invalid assertion	WCSS-1.1-int-2 in ETS for WCSS 1.1
TBC	ETS	Test file [WCSS_Suite] 45.16.08 is poorly written and possibly invalid	WCSS-1.1-con-4 in ETS for WCSS 1.1
TBC	Test Content	Test files for [WCSS_Suite] 45.20.07 are invalid	WCSS-1.1-con-15 in ETS for WCSS 1.1
TBC	Test Content	Test file for [WCSS_Suite] 45.20.22 is invalid	WCSS-1.1-con-15 in ETS for WCSS 1.1
TBC	ETS	Test [WCSS_Suite] 45.26.03 contains an invalid assertion	WCSS-1.1-con-34 in ETS for WCSS 1.1

These PRs need to be raised on the PR system. At time of publishing this report, the Browsing enabler has not been provisioned for the PR system. Full details of the Problem Reports will be found at:
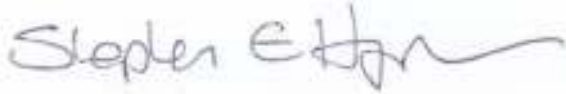http://www.opengroup.org:8000/OMA-PR/

6. Confirmation

This signature states that the included information is true and valid.

_____

Stephen Higgins - Trusted Zone

Appendix A.   Change History   (Informative)

Type of Change	Date	Section	Description