Enabler Validation Plan for Browsing

Approved Version 2.3 – 13 Feb 2009

Open Mobile Alliance

OMA-EVP-Browsing-V2_3-20090213-A

Contents

Figures

Tables

1. Scope

This document details the Validation plan for the Browsing 2.3 Enabler Release. The successful accomplishment of the validation activities will be required for the Enabler to be considered for Approved status.

The validation plan for the Browsing Enabler Release specifications is based on testing expectations in the Enabler Test Requirements (ETR). While the specific test activities to be performed are described in the Enabler Test Specification (ETS) the test environment is described in this plan. This test environment details infrastructure, operational and participation requirements identified for the needed testing activities.

The list of specifications, defining the scope of Browsing 2.3, as stated in [ERELD] and [ETR] is according to the following:

- “WAE Media Types” [WAEMedia]
- ”XHTML Mobile Profile Version 1.1”[XHTMLMP]
- ”XHTML Mobile Profile Version 1.0” [XHTMLMP10]
- ”Wireless Markup Language Version 1.3” [WML1]
- ”Wireless Markup Language Version 2.0” [WML2]
- “ECMAScript Mobile Profile Version 1.0” [ESMP]
- “WMLScript Language” [WMLScript]
- “WMLScript Standard Libraries” [WMLStdLib]
- “WAP Cascading Style Sheet” [WCSS]
- “User Agent Caching Model” [CacheMod]
- “HTTP State Management” [HTTPSM]
- “Wireless Binary eXtended Markup Language” [WBXML]
- ”WMLScript Crypto Library” [CryptoLib]

1.1 Assumptions

None.

1.2 Exclusions

None.

2. References

2.1 Normative References

[IOPPROC] “OMA Interoperability Policy and Process”, Version 1.7, Open Mobile Alliance™, OMA-ORG-IOP_Process-V1_7, URL:http://www.openmobilealliance.org/

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

[ENABLERSPEC] “Enabler Release Definition for Browsing”, Version 2.3, Open Mobile Alliance™, OMA-ERELD-Browsing-V2_3, URL:http://www.openmobilealliance.org/

[IOPETR] “OMA Enabler Test Requirements for Browsing Version 2.3”, Open Mobile Alliance™, OMA-ETR-Browsing-V2_3, URL:http://www.openmobilealliance.org/

[ESMPETS] “OMA Enabler Test Specification for ESMP Version 1.0”. Open Mobile Alliance™, OMA-ETS-ESMP-V1_0-20070824-C, URL:http://www.openmobilealliance.org/

[WCSSETS] “OMA Enabler Test Specification for WCSS Version 1.1”. Open Mobile Alliance™, OMA-ETS-WCSS-V1_1-20070531-C, URL:http://www.openmobilealliance.org/

[WCSSTestCodeETS] “OMA Enabler Test Specification for WCSS Test Code Version 1.1”, Open Mobile Alliance™, OMA-ETS-WCSS_Test_Code-V1_1-20070531-C, URL:http://www.openmobilealliance.org/

[WAEETS] “OMA Enabler Test Specification for WAE Version 2.3”. Open Mobile Alliance™, OMA-ETS-WAE-V2_3-20060509-C, URL:http://www.openmobilealliance.org/

[XHTMLMPCONETS] “OMA Enabler Test Specification for XHTML MP CON Version 1.2”. Open Mobile Alliance™, OMA-ETS-XHTML_MP_CON-V1_2-20080128-C, URL:http://www.openmobilealliance.org/

[ESMPTestCodeETS] “OMA Enabler Test Specification for ESMP Test Code Version 1.0”. Open Mobile Alliance™, OMA-ETS-ESMP_Test_Code-V1_0-20071009-C, URL:http://www.openmobilealliance.org/

[XHTMLMPTestSuitesETS] “OMA Enabler Test Specification for ESMP Test Code Version 1.0”. Open Mobile Alliance™, OMA-ETS-XHTML_MP_Test_Suites-V1_2-20080128-C, URL:http://www.openmobilealliance.org/

[IOPEICSClient] “OMA Enabler Implementation Compliance Statement for Browsing Version 2.3”. Open Mobile Alliance™, OMA-EICS-Browsing_Client-V2_3, URL:http://www.openmobilealliance.org/

[IOPEICSServer] “OMA Enabler Implementation Compliance Statement for Browsing Version 2.3”. Open Mobile Alliance™, OMA-EICS-Browsing_Server-V2_3, URL:http://www.openmobilealliance.org/

[CacheMod] “WAP Caching Model”, Open Mobile Alliance™. WAP-120-UACach. URL:http://www.openmobilealliance.org/

[CacheOp] “WAP Cache Operation”, Open Mobile Alliance™. WAP-175-CacheOp. URL:http://www.openmobilealliance.org/

[CryptoLib] “WMLScript Crypto Library Specification”, Open Mobile Alliance™. WAP-161-WMLScriptCrypto. URL:http://www.openmobilealliance.org/

[DLARCH] “Download Architecture”. Open Mobile Alliance^{TM}. OMA-Download-ARCH-v1_0. URL:http://www.openmobilealliance.org

[DLOTA] “Generic Content Download Over The Air Specification”, Open Mobile Alliance™. OMA-Download-OTA-v1_0. URL:http://www.openmobilealliance.org/

[DRM] “Digital Rights Management”, Open Mobile Alliance™. OMA-Download-DRM-v1_0. URL:http://www.openmobilealliance.org/

[DRMCF] “DRM Content Format”, Open Mobile Alliance™. OMA-Download-DRMCF-v1_0. URL:http://www.openmobilealliance.org/

[DRMREL] “DRM Rights Expression Language”, Open Mobile Alliance™. OMA-Download-DRMREL-v1_0. URL:http://www.openmobilealliance.org/

[EFI] “External Functionality Interface Framework”, Open Mobile Alliance™. WAP-231-EFI. URL:http://www.openmobilealliance.org/

[ESMP] “ECMAScript Mobile Profile”, Open Mobile Alliance™. OMA-WAP-ESMP-V1_0. URL:http://www.openmobilealliance.org/

[ESMPCrypto] “ECMAScript Crypto”, Open Mobile Alliance™. OMA-WAP-ECMACR-V1.0. URL:http://www.openmobilealliance.org/

[HTTPSM] “HTTP State Management Specification”, Open Mobile Alliance™. WAP-223-HTTPSM. URL:http://www.openmobilealliance.org/

[PICT] “WAP Pictogram”, Open Mobile Alliance™. WAP-213-WAPInterPic. URL:http://www.openmobilealliance.org/

[PSTOR] “WAP WAG Persistent Storage Interface”, Open Mobile Alliance™. WAP-301-PSTOR. URL:http://www.openmobilealliance.org/

[RFC2109] “HTTP State Management Mechanism”, D. Kristol and L. Montulli, February 1997. URL: http:/www.ietf.org/rfc/rfc2109.txt

[RFC2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997. URL:http://www.ietf.org/rfc/rfc2234.txt

[RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et al, June 1999. URL: http://www.ietf.org/rfc/rfc2616.txt

[WAE] “Wireless Application Environment Specification”, Open Mobile Alliance™. OMA-WAP-WAESpec-V2_1. URL:http://www.openmobilealliance.org/

[WAEMedia] “WAE Defined Media Type”, Open Mobile Alliance™. WAP-237-WAEMT. URL:http://www.openmobilealliance.org/

WAPTLS ”WAP TLS Profile and Tunneling”, Open Mobile Alliance™. WAP-219-TLS. URL:http://www.openmobilealliance.org/

[WBXML] “WAP Binary XML Content Format”, Open Mobile Alliance™. WAP-192-WBXML. URL:http://www.openmobilealliance.org/

[WCSS] “WAP CSS”, Open Mobile Alliance™. WAP-239-WCSS. URL:http://www.openmobilealliance.org/

[WML1] “Wireless Markup Language Version 1.3”, Open Mobile Alliance™. WAP-191-WML. URL:http://www.openmobilealliance.org/

[WML2] “Wireless Markup Language”, Open Mobile Alliance™. WAP-238-WML. URL:http://www.openmobilealliance.org/

[WMLScript] “WMLScript Language Specification”, Open Mobile Alliance™. WAP-193-WMLS. URL:http://www.openmobilealliance.org/

[WMLStdLib] “WMLScript Standard Libraries Specification”, Open Mobile Alliance™. WAP-194-WMLSL. URL:http://www.openmobilealliance.org/

[WTAI] “Wireless Telephony Application Interface Specification”, Open Mobile Alliance™. WAP-268-WTAI. URL:http://www.openmobilealliance.org/

[XHTMLMP] “XHTML Mobile Profile 1.1”, Open Mobile Alliance™. OMA-WAP--XHTMLMP-V1_1.. URL:http://www.openmobilealliance.org/

[XHTMLMP10] “XHTML Mobile Profile”, Open Mobile Alliance™. WAP-277-XHTMLMP. URL:http://www.openmobilealliance.org/

[WTA] “Wireless Telephony Application Specification”, Open Mobile Alliance™. WAP-266-WTA. URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMADICT] “Dictionary for OMA Specifications”, Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:http://www.openmobilealliance.org/

[CSS2] “Cascading Style Sheets, level 2 (CSS2) Specification”, W3C Recommendation, Bert Bos et al., 12 May 1998. URL:http://www.w3.org/TR/1998/REC-CSS2-19980512.

[CSSMP] “CSS Mobile Profile 1.0”, W3C Candidate Recommendation, Ted Wugofski. Doug Dominiak, Peter Stark, 24 October 2001. URL:http://www.w3.org/TR/2001/CR-css-mobile-20011024.

[ECMA327] Standard ECMA-327, “ECMAScript 3rd Edition Compact Profile”, ECMA, June 2001, URL: ftp://ftp.ecma.ch/ecma-st/Ecma-327.pdf

[ECMAScript] Standard ECMA-262: “ECMAScript Language Specification – Edition 3”, ECMA, December 1999. URL: ftp://ftp.ecma.ch/ecma-st/Ecma-262.pdf

[ESMP_Suite] OMA compliancy test suite for EcmaScript. http://member.openmobilealliance.org/ftp/iop/iop-Browsing/gen_info/IOP-Browsing-Test-Suite.shtml

[XHTML_Suites] OMA compliancy test suite for XHTML. http://member.openmobilealliance.org/ftp/iop/iop-Browsing/gen_info/IOP-Browsing-Test-Suite.shtml

[ProvArch] “WAP Provisioning Architecture Overview”, Open Mobile Alliance™. WAP-182-ProvArch. URL:http://www.openmobilealliance.org/

[ProvCont] “WAP Provisioning Content Specification”, Open Mobile Alliance™. WAP-183-ProvCont. URL:http://www.openmobilealliance.org/

[ProvUAB] “WAP Provisioning User Agent Behaviour Specification”, Open Mobile Alliance™. WAP-185-ProvUAB. URL:http://www.openmobilealliance.org/

[UAPROF] “WAG UAProf”, Open Mobile Alliance™. WAP-248-UAPROF. URL:http://www.openmobilealliance.org/

[VSWAP 2.0] OMA compliancy test suite for WAP 2.0 testing URL: http://member.openmobilealliance.org/ftp/iop/iop-Browsing/gen_info/IOP-Browsing-Test-Suite.shtml

[WAE20] “Wireless Application Environment Specification – version 2.0”, Open Mobile Alliance™. WAP-236-WAESpec. URL:http://www.openmobilealliance.org/

[WAPArch] “WAP Architecture Specification”, Open Mobile Alliance™. WAP-210-WAPArch. URL:http://www.openmobilealliance.org/

[WCSS_Suite] OMA compliancy test suite for WCSS testing URL: http://member.openmobilealliance.org/ftp/iop/iop-Browsing/gen_info/IOP-Browsing-Test-Suite.shtml

WDP “Wireless Datagram Protocol”, Open Mobile Alliance™. WAP-259-WDP. URL:http://www.openmobilealliance.org/

[W-HTTP] “WAP Wireless Profiled HTTP”, Open Mobile Alliance™. WAP-229-HTTP. URL:http://www.openmobilealliance.org/

[WSP] “Wireless Session Protocol”, Open Mobile Alliance™. WAP-230-WSP. URL:http://www.openmobilealliance.org/

W-TCP ”WAP Wireless Profile TCP”, Open Mobile Alliance™. WAP-225-TCP. URL:http://www.openmobilealliance.org/

WTLS ”Wireless Transport Layer Security”, Open Mobile Alliance™. WAP-261-WTLS. URL:http://www.openmobilealliance.org/

[WTP] “Wireless Transaction Protocol Specification”, Open Mobile Alliance™. WAP-224-WTP. URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

All sections and appendixes, except “Scope”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Application Addressing the ability to address a particular user agent on a WAP client

Browsing The act of using a **PC Browser** or **WAE User Agent** to access content from a server using the established WWW request-response model using HTTP to establish sessions between client and server through intermediate networks and performance enhancing proxies. **Client** - in the context of push, a client is a device (or service) that expects to receive push content from a server. In the context of pull a client, it is a device initiates a request to a server for content or data. See also "device".

Contact Point address information that describes how to reach a push proxy gateway, including transport protocol address and port of the push proxy gateway.

Content subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent on a client. Content can either be returned in response to a user request, or pushed directly to a client.

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

Content Format (or Format) actual representation of content

Deprecated A deprecated feature (e.g. specification, element or attribute) is one that has been outdated by a newer feature. Deprecated features are defined in the specification and are clearly marked as deprecated. Deprecated features may become obsolete in a future version.

Device a network entity that is capable of sending and receiving packets of information and has a unique device address. A device can act as both a client and a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server

ECMA Script a scripting language produced and managed by the European Computer Manufacturers Association (ECMA) which provides a common scripting language for the computer industry

Enabler Release a collection of specifications that combined together form an enabler for a service area, e.g. a download enabler, a browsing enabler, a messaging enabler, a location enabler, etc. The specifications that are forming an enabler should combined fulfill a number of related market requirements

End-user see "user"

Gateway (or WAP Gateway) a server which acts as an intermediary for some other server. A gateway performs protocol transformation as well as encoding/decoding content.

Host Object ECMA Script objects provided by the user agent for the purpose of interaction with the loaded document.

Hypermedia Transfer The hypermedia transfer services provides for the transfer of self-describing hypermedia resources. The combination of WSP (Wireless Session Protocol) [WSP] and WTP (Wireless Transaction Protocol) [WTP] provide the hypermedia transfer service over secure and non-secure datagram transports over datagram-based protocol stack. The W-HTTP (Wireless Profiled Hypertext Transfer Protocol) [W-HTTP], a profile of HTTP/1.1 [[RFC2616] provides the hypermedia transfer service over secure and non-secure connection-oriented transports over connection-oriented protocol stack.

Origin Server the server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

Media type a MIME media type or an identifier for a given data type.

Minimum Functionality Description Description of the guaranteed features and functionality that will be enabled by implementing the minimum mandatory part of the Enabler Release.

PC Browser an existing Web browser that supports text/html.

Resource a network data object or service that can be identified by a URL. Resources may be available in multiple representations (e.g., multiple languages, data formats, size, and resolutions) or vary in other ways.

Server a device (or application) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client. Also Origin Server.

Terminal a device typically used by a user to request and receive information. Also called a mobile terminal or mobile station.

Terminal-ID an identifier that is used by a PPG to uniquely identify a terminal.

User a user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content. Also referred to as end-user.

User agent a user agent (or content interpreter) is any software or device that interprets resources. This may include textual browsers, voice browsers, search engines, etc.

WAE User Agent (or User Agent) a User Agent is any software or device that interprets markup and scripting languages or other content. This may include textual browsers, voice browsers, search engines, etc.

WAE version the version of the WAE User Agent. The version of the WAE User Agent may be uniquely identifiable by the WAP version, e.g. WAP version 1.1 contains WAE version 1.1, or it may be a feature of the WAP version in which case the WAE versioning mechanisms are used to determine the WAE User Agent version.

WAP1 WAP Version 1, nominally the latest point release, e.g. WAP V1.2.1, unless otherwise noted.

WAP2 WAP Version 2. When used as a prefix, it indicates that something is compliant to the WAP Version 2 conformance requirements, e.g., a WAP2 client is a client that fulfils all the requirements for a user agent of WAP Version 2. WAP2 content is content with a media type specified in WAP Version 2.

WAP Proxy an intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible translation, to other servers. It may provide functions of protocol enhancement, transcoding or any number of other optimization or transformation functions and may be associated with any gateways, proxies or servers being used in the deployment architecture.

	WAP gateway is one of the optional functionalities of WAP proxy.
WML	The Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrowband device, e.g., a phone.
WML Script	A scripting language used to program the mobile device. WML Script is modeled on the ECMA Script scripting language.
XHTML	The W3Cs codification of HTML version 4.01 in an XML.
XML	the Extensible Markup Language is a World Wide Web Consortium (W3C) standard for Internet markup languages, of which WML is one such language. XML is a restricted subset of SGML.
vCalendar	Internet Mail Consortium (IMC) electronic calendar record.
vCard	Internet Mail Consortium (IMC) electronic business card.

3.3 Abbreviations

CSS	Cascading Style Sheets
DRM	Digital Rights Management
ECMA	European Computer Manufacturer Association
EFI	External Functionality Interface
ERDEF	Enabler Requirement Definition
ERELD	Enabler Release Definition
ESMP	ECMAScript Mobile Profile see [ESMP]
ETR	Enabler Test Requirements, see [IOPProc]
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPSM	HypeText Transfer Protocol State Management, see [HTTPSM]
OMA	Open Mobile Alliance
OTA	Over The Air
OTA-HTTP	(Push) OTA over HTTP
OTA-HTTP-TLS	OTA-HTTP over TLS
OTA-WSP	(Push) OTA over WSP
PDP	Packet Data Protocol
PI	Push Initiator
PO-TCP	PPG Originated TCP connection establishment method
PPG	Push Proxy Gateway
RTT	Round Trip Time
SI	Service Indication
SL	Service Loading
TOD	Time Of Day
WAESpec	Wireless Application Environment Specification, see [WAESpec]
WAP	Wireless Application Protocol
WCSS	Wireless Cascading Style Sheet, see [WCSS]
UAProf	User Agent Profile
W3C	World Wide Web Consortium
W-HTTP	Wireless Profiled HTTP
WML	Wireless Markup Language (WML1 or WML2)
WML1	Wireless Markup Language Version 1.3
WML2	Wireless Markup Language Version 2.0
WWW	World Wide Web

WSP	Wireless Session Protocol
WAP	Wireless Application Protocol
WAE	Wireless Application Environment. Unless otherwise stated it refers to this version.
WAE20	Wireless Application Environment version 2.0 [WAE20]
WTA	Wireless Telephony Application
WTAI	Wireless Telephony Application Interface
WBMP	Wireless BitMaP
XHTML	Extensible HyperText Markup Language
XHTMLMP	Extensible HyperText Markup Language Mobile Profile, see [XHTMLMP]

4. Enabler Validation Description

See section 5.

5. TestFest Activities

5.1 Enabler Test Guidelines

A full description of Browsing V2.3 can be found in the ERELD and specifications.

Browsing 2.3 provides the OMA browsing capability for mobile and wireless handheld devices by defining the necessary capabilities of client and server in the end to end request-response model. Browsing 2.3 also specifies any necessary or optional supporting network services that may be provided on a gateway or proxy.

Browsing 2.3 uses much of the internet technology used in today’s PC Browsers to access content on the World Wide Web (WWW) but limits the specified profiles of this technology to that appropriate to the constrained resources and user interface of mobile and wireless handheld devices, e.g. reduced memory, processing power, communications bandwidth, display and user input capabilities, including some extensions to improve the user experience.

Browsing V2.3 is an enhancement of Browsing V2.2, providing the markup and script convergence with the Internet, through the use of XHTML Mobile Profile [XHTMLMP], Wireless Cascading Style Sheets [WCSS] and ECMAScript Mobile Profile [ESMP]. Browsing 2.3 differs from Browsing V2.2 through an updated version of [XHTML].

The suite of specifications defining Browsing V2.3 defines the application-level protocols, semantics, syntax, content formats, user agent behavior, and the use of hypermedia transfer protocols required to achieve consistent function and interoperability of services.

Browsing 2.3, or the WAE User Agent, support the following features directly through the WAE Specification [WAESpec]:

- Markup language based content to be rendered to the user of the device;
    - WML V1.3 [WML1], WML V2.0 [WML2], XHTMLMP V1.0 [XHTMLMP10] and XHTMLMP1.1 [XHTMLMP] are specified. The XHTMLMP specification also provides HTML rendering capability within the limit of the device’s capabilities.
- Scripting language augmentation of the markup content to allow extended functionality and user experience;
    - WMLScript [WMLScript], with its associated WMLScript Library [WMLStdLib], and
    - ECMAScript Mobile Profile [ESMP] along with a well known set of host objects.
- Style capabilities to enhance the presentation of markup on devices supporting it.
    - The style is provided by the WAP Cascading Style Sheet [WCSS] specification which is an profile of the W3C’s CSS2 [CSS2] and being inspired by the CSS2 mobile profile [CSSMP] extended with some other desirable features not available in [CSS2] to suite the needs of the mobile Browsing environment.
- Image and other content support;
    - WBMP is a unique, efficient, monochrome format for devices and predecessor devices but other types are supported, the types dependent on the device. WBMP is defined in the WAE Media Types specification [WAEMedia].
    - Vcard [VCARD] and Vcalendar[VCAL] are supported formats for the exchange of electronic business cards and calendar information
- Local caching of content to improve user experience and reduce network usage.[CacheMod];
- HTTP State Management [HTTPSM], or cookies in common terminology, to provide the means to convey state and state information between user and application server, e.g. session identifiers, time and date information of last access, recent enquiries to that application, to aid the user’s access to that application;
- Pictograms [PICT] to provide an enhanced user experience through the use of small images to augment or even replace textural information, e.g. the use of common weather symbols to illustrate the current weather; and in which combinations and whether they are mandatory or optional.

The Browsing 2.3 enabler also supports optional extensions to this basic browsing environment, namely

- Download and DRM ERELD (comprising [DLARCH], [DLOTA], [DRM], [DRMREL] & [DRMCF])

  - Provides a common means to download content over the air and manage the lifecycle of the content using Digital Rights Management with the rights expressed in a Rights Expression Language.

- External Functionality Interface [EFI]
  - EFI extends the browser to include other hardware or software elements through the use of markup and script interfaces. The discovery of these elements is enabled thereby allowing them to be used, e.g. start or stop another application, retrieve a digital photograph from a camera, etc.
- Push ([PushArch], [CacheOp], [PushMessage], [PushOTA], [PAP], [PPGService], [ServiceInd] [ServiceLoad])
  - PUSH provides an alert mechanism with the ability to have the alert provide a link to content which is subsequently pulled using the browser
- MMS ERELD [MMS-ERELD]
  - The Multimedia Messaging Service provides the means to send and receive rich media messages and uses Push for the alert mechanism.
- Synchronization [Sync]
- Application level signing of content
  - This is provided through the use of scripting extensions of the basic scripting environment by the ECMA Script Cypto Object [ESMPCrypto] and WMLScript Crypto Library [CryptoLib] features
- WBXML [WBXML]
  - A compact format used for WAP Version 1.x browsers and still supported for other features though not required for the Browser per se.
- Wireless Telephony Application Interface [WTAI]
  - The WTA [WTA] feature is not specifically addressed by the Browsing enabler though it utilizes many of the Browsing features, but the Browsing specification does provides access to the Public Wireless Telephony Application Interface (WTAI) [WTAI].
- Persistent Storage [PSTOR]
  - This provides a means to store data objects locally, personal details, applications, etc., within a device for use by applications and allows improved user experience

Features such as Provisioning, [ProvArch], [ProvCont] and [ProvUAB], and UAPROF [UAPROF] are not specifically called out but are related and presented in the [WAE].

5.1.1 Minimal Test Configuration

The minimal test configuration of [Browsing 23] shall include:

- A Client implementation
- A Server implementation
- A Content server providing necessary test suites that is accessible through a mobile telephony network
- A Mobile telephony network

5.1.2 Minimal Participation Guidelines

Minimum

- 3 client implementations and

3 WAP GW/Proxies are needed

5.1.3 Optimal TestFest Achievement Guidelines

The ETS Test Cases listed below represent a subset of all the Test Cases for the Enabler that it is thought can be executed in a test session at an OMA TestFest. This list is intended to facilitate maximum test coverage of the functionality of the enabler within a test session. It is not intended to be the only tests executed at a TestFest, and teams are encouraged to execute more tests if they are able to do in the time allowed.

Client settings shall be in accordance with the network parameters provided by the TestFest host.

- Gateway and proxy configuration shall be in accordance with the information provided by the TestFest host for serving all clients participating in the TestFest.
- Origin server configuration shall be in accordance with the configuration information provided by the TestFest host for providing content to all clients participating in the TestFest.
- The SMS-C configuration shall provide the agreed interface to the gateways and SMS service to the clients participating in the TestFest.

Configuration of the AP and the RADIUS shall provide RADIUS ACCOUNTING to the gateways/proxies requiring it for MSISDN forwarding. Necessary configuration parameters shall be in advance documented.

5.2 Enabler Test Requirements

5.2.1 Test Infrastructure Requirements

The testing shall be performed as end-to-end testing. Most likely the client participants will be in one place, while the participant gateways and proxies will be located in member companies premises, accessible to the rest of the test fest environment. Such a “distributed” test fest environment puts effort on the test fest host and requires detailed documented configuration.

The Network Elements involved in Browsing Testing are:

- PLMN (GSM/GPRS)
- An SMS-C Supporting SMPP 3.4 or higher
- A Gateway supporting Browsing 2.3
- Clients supporting Browsing 2.3
- Origin servers holding the Browsing test content

A Network Analyzer for Network Monitoring/Protocol Analyzing is also useful

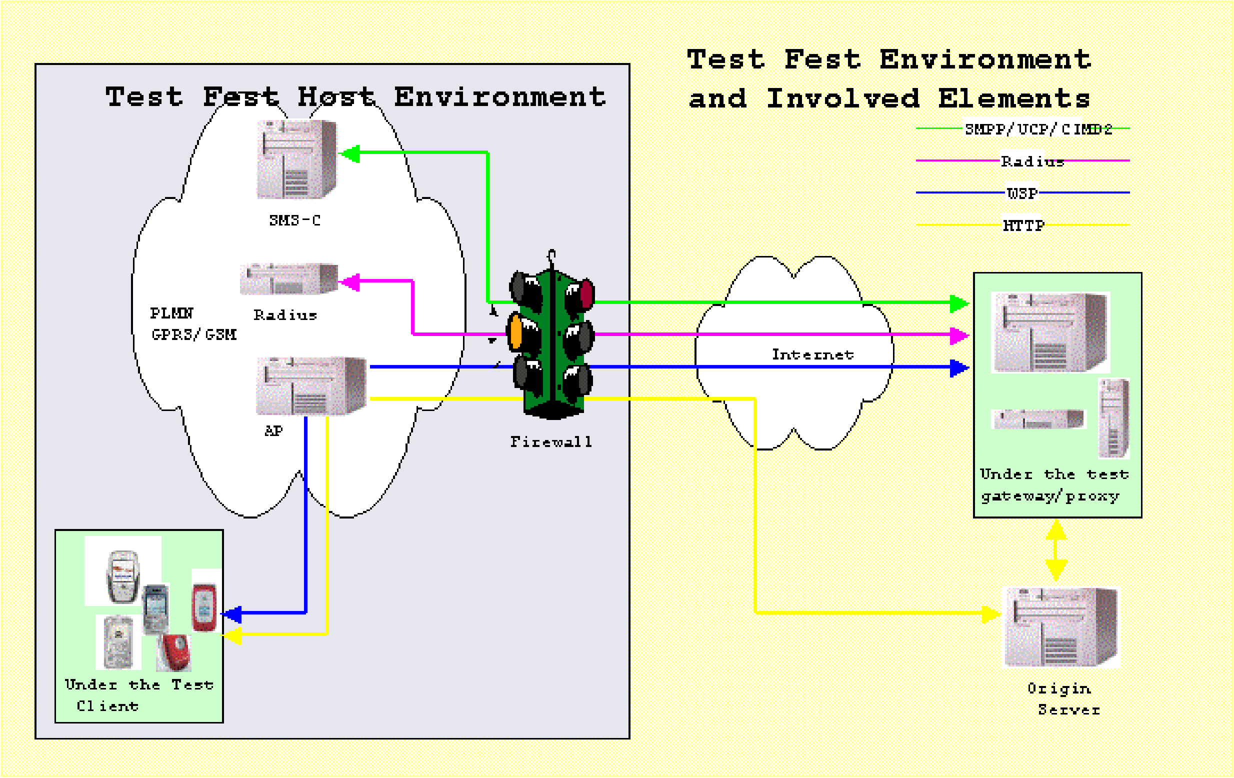

Figure 1: Involved elements for testing browsing

5.2.2 Enabler Execution Flow

The following diagrams provide a high level overview of the message exchanges between the involved elements of the Browsing test environment based on the type of bearers, i.e. WAP or WHTTP stack.

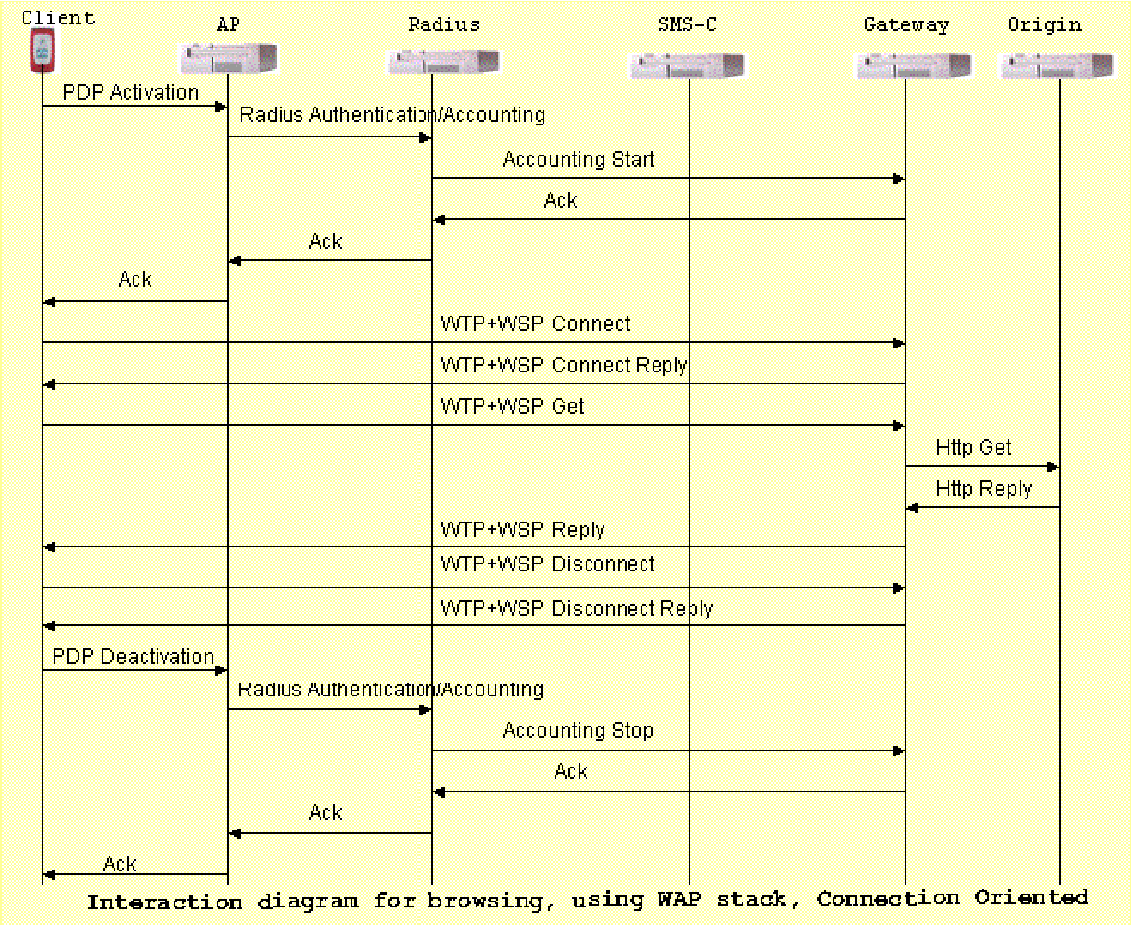

Figure 2 : Connection Oriented, WAP Stack

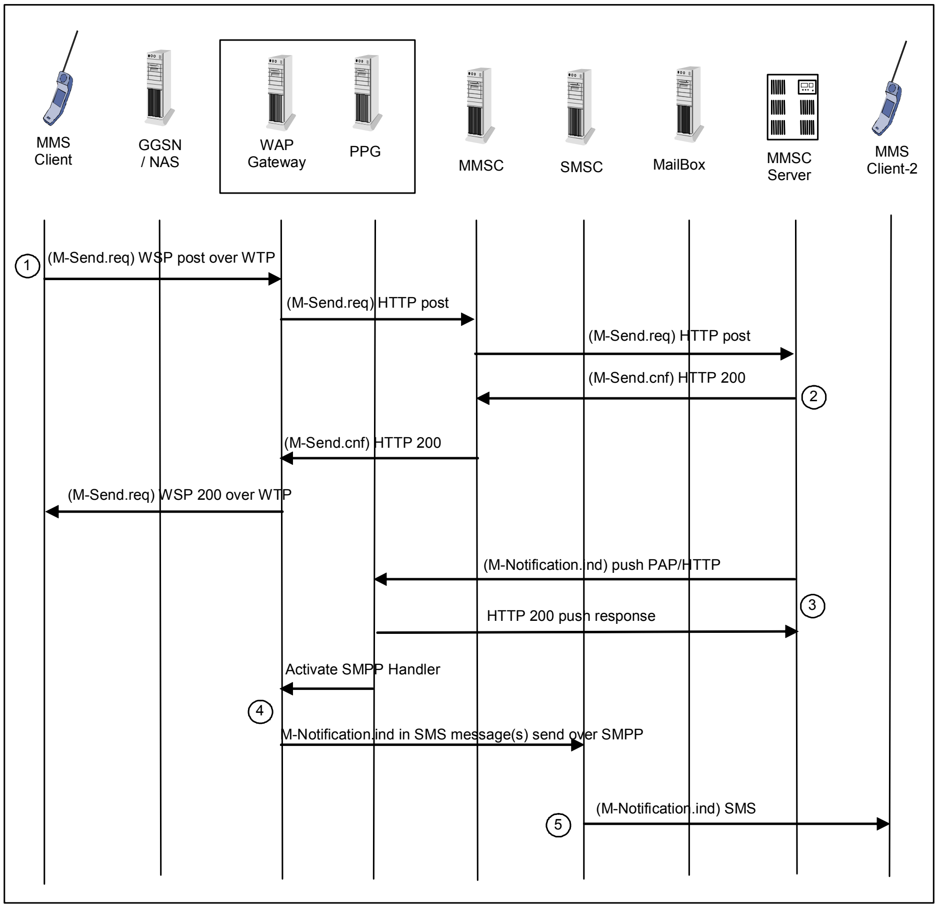

Figure 3 : Example Call Flow

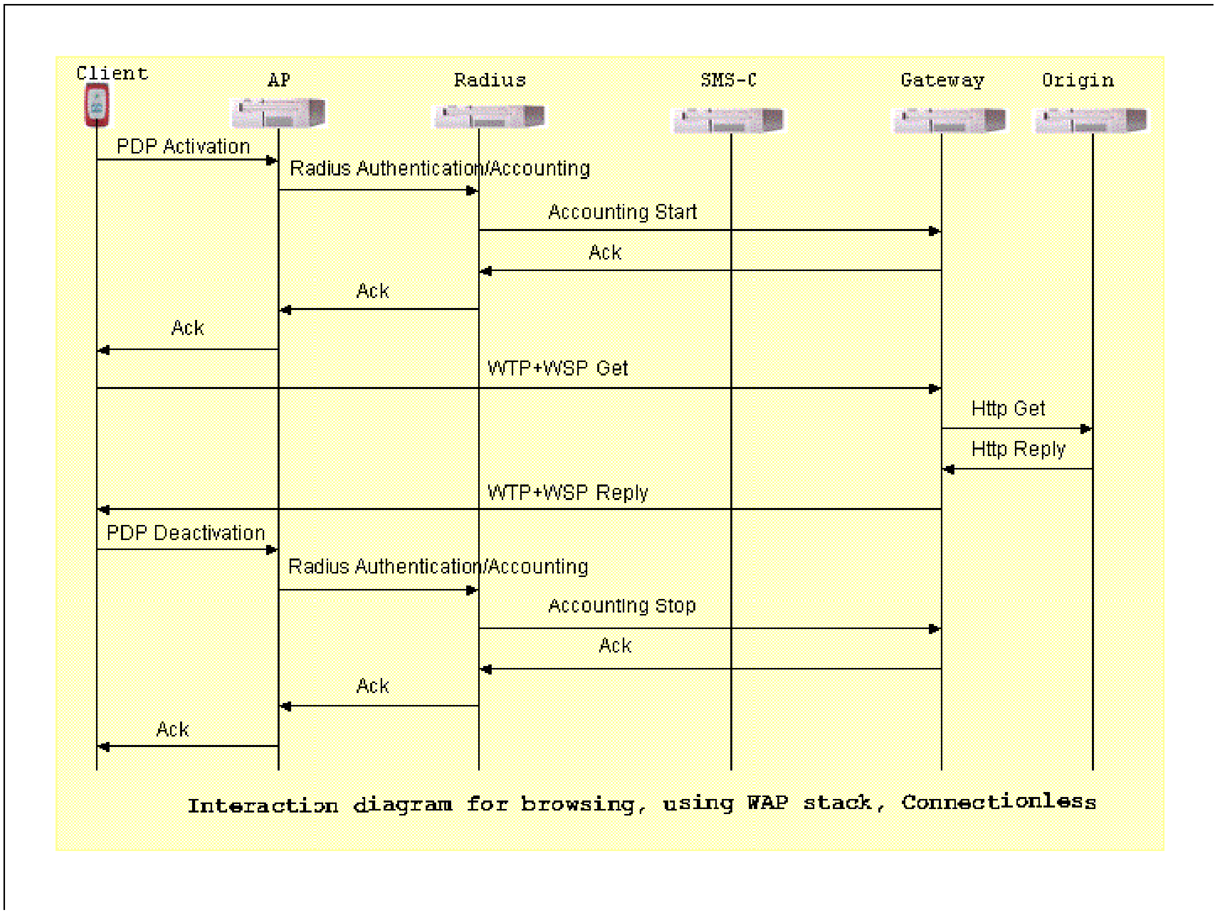

Figure 4 : Conectionless, WAP Stack

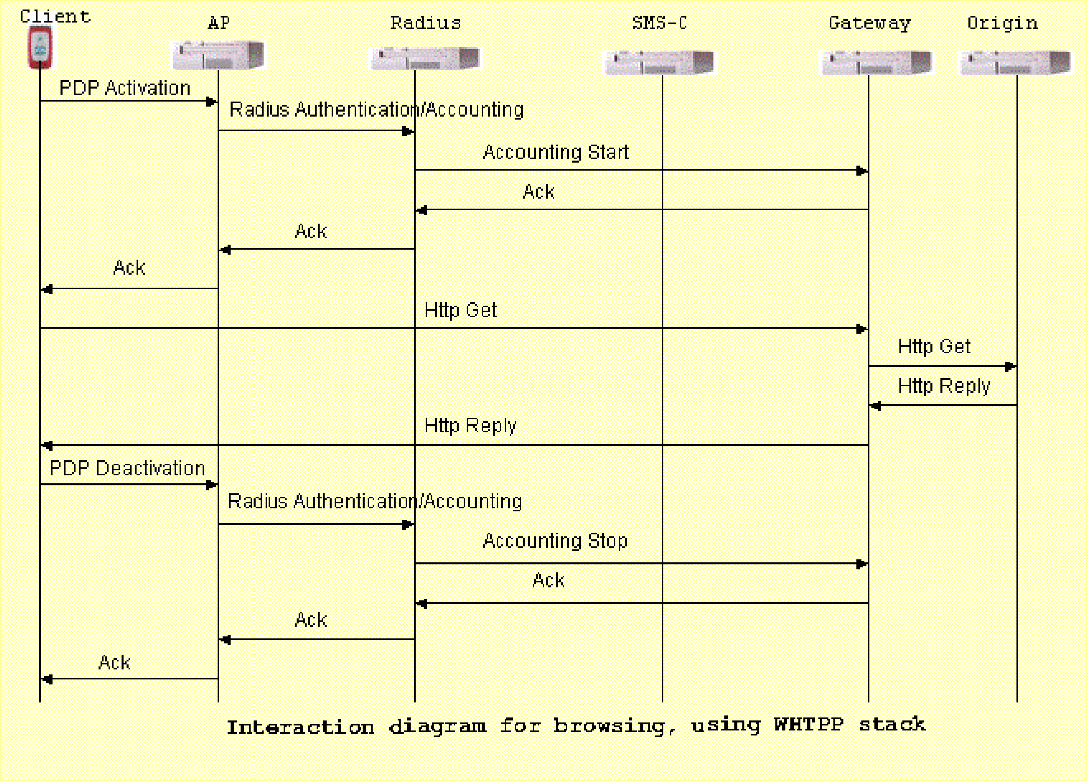

Figure 5 : WHTTP Stack

5.2.3 Test Content Requirements

The test content at the OMA Content Server (http://testfest.openmobilealliance.org/) shall be used.

5.2.4 Test Limitations

5.2.4.1 Physical

None.

5.2.4.2 Resources

None

5.2.5 Test Restrictions

5.2.5.1 Test Session Entrance Criteria

The test session entry criteria are defined by the TestFest Participation Guidelines of the IOP WG.

5.2.5.2 Technical Prerequisites

- Client settings shall be in accordance with the network parameters provided by the test fest host.

- Gateway and proxy configuration shall be in accordance with the information provided by the test fest host for serving all clients participating in the test fest.
- Origin server configuration shall be in accordance with the configuration information provided by the test fest host for providing content to all clients participating in the test fest.
- The SMS-C configuration shall provide the agreed interface to the gateways and SMS service to the clients participating in the test fest.
- Configuration of the AP and the RADIUS shall provide RADIUS ACCOUNTING to the gateways/proxies requiring it for MSISDN forwarding. Necessary configuration parameters shall be in advance documented.

5.2.6 Test Tools

5.2.6.1 Existing Tools to be Used

The test tools needed to carry out the conformance and interoperability test cases are listed as:

- VSWAP 2.0
- ESMP_Suite
- WCCS_Suite
- XHTML_Suites

5.2.6.2 Test Tool Requirements

None.

5.2.7 Resources Required

All test cases can be run in a 3 hour slot. However, comparative testing may require more time, depending on the number of devices being compared.

5.3 Tests to be Performed

The following sections describe the tests related to the formal TestFest validation activities.

5.3.1 Entry Criteria for TestFest

The following tests need to be performed and passed by implementations by members wishing to participate in the TestFest. This ensures minimal requisite capability of the implementations. The tests are defined in the ETS [XXXETS] and any special comments are noted.

The test session entry criteria are defined by the TestFest Participation Guidelines of the IOP WG.

Client settings shall be in accordance with the network parameters provided by the TestFest host.

- Gateway and proxy configuration shall be in accordance with the information provided by the TestFest host for serving all clients participating in the TestFest.

- Origin server configuration shall be in accordance with the configuration information provided by the TestFest host for providing content to all clients participating in the TestFest.

- The SMS-C configuration shall provide the agreed interface to the gateways and SMS service to the clients participating in the TestFest.

- Configuration of the AP and the RADIUS shall provide RADIUS ACCOUNTING to the gateways/proxies requiring it for MSISDN forwarding. Necessary configuration parameters shall be in advance documented.

5.3.2 Testing to be Performed at TestFest

The following tests need to be performed to fully cover the range of capabilities of the enabler and defined protocols. These tests are to be covered in the TestFest. The tests are defined in the ETS [XXXETS] and any special comments are noted.

Test Case Id	Special Conditions
ESMP-1.0-conf-2	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-3	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-4	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-5	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-7	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-8	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-9	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-10	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-11	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-12	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-13	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-14	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-15	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-16	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-17	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-18	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-19	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-20	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-21	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-22	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-23	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-24	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-25	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-26	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-27	Test environment is set up and Client devices are configured according to [BrowsingETP]

Test Case Id	Special Conditions
ESMP-1.0-conf-28	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-29	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-30	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-31	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-32	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-33	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-34	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-35	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-36	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-37	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-38	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-39	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-41	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-42	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-43	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-45	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-46	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-47	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-48	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-49	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-50	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-51	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-52	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-53	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-55	Test environment is set up and Client devices are configured according to [BrowsingETP]

Test Case Id	Special Conditions
ESMP-1.0-conf-56	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-57	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-58	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-59	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-62	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-63	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-64	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-65	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-66	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-67	Test environment is set up and Client devices are configured according to [BrowsingETP]
ESMP-1.0-conf-68	Test environment is set up and Client devices are configured according to [BrowsingETP]
WCSS-1.1-con-1	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-2	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-3	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-4	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-5	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-6	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-7	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-8	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-9	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-10	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-11	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-12	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-13	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-14	Test environment is set up and Client devices are configured according to [WCSS_Env]

Test Case Id	Special Conditions
WCSS-1.1-con-15	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-16	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-17	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-18	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-19	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-20	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-21	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-22	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-23	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-24	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-25	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-26	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-27	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-28	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-29	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-30	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-31	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-32	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-conf-33	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-co-34	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-35	Test environment is set up and Client devices are configured according to [WCSS_Env]
WCSS-1.1-con-36	Test environment is set up and Client devices are configured according to [WCSS_Env]
xHTML-1.2-con-1	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-2	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-3	Test environment is set up and Client device configured according to [TestEnv]

Test Case Id	Special Conditions
xHTML-1.2-con-4	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-5	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-6	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-7	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-8	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-9	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-10	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-11	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-12	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-13	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-14	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-15	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-16	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-17	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-18	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-19	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-20	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-21	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-22	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-23	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-24	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-25	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-26	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-27	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-28	Test environment is set up and Client device configured according to [TestEnv]

Test Case Id	Special Conditions
xHTML-1.2-con-29	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-30	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-31	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-32	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-33	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-34	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-35	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-36	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-37	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-38	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-39	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-40	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-41	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-42	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-43	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-43	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-44	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-45	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-46	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-47	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-48	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-49	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-50	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-51	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-52	Test environment is set up and Client device configured according to [TestEnv]

Test Case Id	Special Conditions
xHTML-1.2-con-53	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-54	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-55	Test environment is set up and Client device configured according to [TestEnv]
xHTML-1.2-con-56	Test environment is set up and Client device configured according to [TestEnv]

Table 1: Listing of Tests to be Performed at TestFest

5.4 Enabler Test Reporting

5.4.1 Problem Reporting Requirements

Normal Reporting, no special reporting required.

5.4.2 Enabler Test Requirements

Normal Reporting, no special reporting required

6. Alternative Validation Activities

None.

7. Approval Criteria

The Browsing 2.3 Enabler can be put in the Approved state when:

- The Enabler has been tested successfully at 3 Test Fests or
- 5 companies have successfully run bi(Uni-)lateral tests towards the OMA test server and has reported results and
- No open PRs exist.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-EVP-Browsing-V2_3	13 Feb 2009	TP Approved TP#24 Macau

Appendix B. Testing Requirements

B.1 ESMP

B.1.1 User Agent (Client)

B.1.1.1 Deployment

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the User Agent is able handle XHTML content type documents with embedded ECMAScript.	The device must be able to receive script content and deliver it to the browser user agent. The user agent MUST understand that this is script content and handle it as such. Both positive and negative tests should be attempted. These tests should be insensitive to the existence of any intermediate proxies. The testing of the <script> and <noscript> elements are key here	[ESMP_Suite] TCs from chapter 7.2, namely 49.19.02-10 (9 TCs)
The test campaign needs to verify the User Agent to will respect the domain security of the ESMP specification	Tests should be written that attempt to access network content that is not within the browsers accepted domain. This will involve using the location object.	[ESMP_Suite], TCs from chapter 9.5.2.2, namely 49.29.08 (1 TCs)
The test campaign needs to verify the User Agent to will respect the history constraints of the ESMP specification	Tests should be written that attempt to breach the history stack constraints, as defined by the [ESMP] specification. This will involve using the 'history' object.	[ESMP_Suite], TCs from chapter 9.3.3, namely 49.27.07,09 and a new one for the go() method (3 TCs)
The test campaign needs to verify the User Agent can handle ECMAScript Language syntax definition as defined in ESMP	A general conformance testing for language syntax should be carried out. This should consist of touch testing each of the ECMA-262 objects as well as the context objects. Many ECMAScript and Javascript test suites exist. A survey should be done to determine where to best obtain this type of testing material. As a conformance test, this work should be conducted prior to any interoperability testing, and may be a good candidate for so-called self certification.	[ESMP_Suite] All the TCs from chapter 5, missing TCs for the Variable and Data Types and Type conversions (139 TCs)

Table 2 ESMP User Agent Deployment Requirements

B.1.1.2 Semantic Interoperability

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the User Agent to correctly supports the language semantics of ESMP	All devices supporting [ESMP] should respond consistently when presented the same scripts. A series of scripts that generate documented outcomes should be assembled and applied to devices. The results and side-effects of the test scripts should be compared to other devices to guarantee consistency. Script content should broadly touch all ESMP defined objects, and should be built in an extensible manner, so that discovered areas of inconsistency can be added to the suite of tests.	[ESMP_Suite] Some TCs from chapter 6, proposed one from each object suite (8 TCs)
The test campaign needs	All ESMP language objects and host object should be touch	[ESMP_Suite] Some TCs

to verify the User Agent to correctly handles object semantics defined in [ESMP]	tested. These tests may be combined with the above listed language semantic tests.	from chapter 9, proposed one per object (5 TCs)

Table 3 ESMP User Agent Semantic Interoperability Requirements

B.1.1.3 Processing model Interoperability

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the User Agent to correctly handles all of the invocation approaches mechanisms as defined in ESMP	All of the invocation modes and exits for script must be tested.	[ESMP_Suite] The TCs from section 7.2 that test the 4 invocation modes (4 TCs)
The test campaign needs to verify the User Agent to responds to error conditions in an interoperable way	Runtime errors must be tested to insure that both fatal and non-fatal errors are handled in a consistent, and interoperable way. Of special note are tests that should cover infinite loops and tests that should cover memory exhaustion.	[ESMP_Suite] The TCs from section 6.9.4 New TCs have to be written to test the infinite loops and memory exhaustion (9 TCs 2 new)

Table 4 ESMP Use Agent Deployment requirements

B.1.2 Priorities in Interoperability

SUMMARY REQUIREMENT	PRIORITY	TEST TOOL
ECMAScript Type Support	High	[ESMP_Suite] The TCs from chapter 7
Semicolon support at end of statements	High	[ESMP_Suite] The TC from section 5.2, 49.01.10
Support for EvalError Exception	High	[ESMP_Suite] The TC from section 6.1.3.10, 49.06.36
Native Object Support – String Object	High	[ESMP_Suite] The TCs from section 6.3
Native Object Support – Error Object	High	[ESMP_Suite] The TCs from section 6.9
Support for DOM2 compliant event binding	High	[ESMP_Suite] The TCs from section 8.2
Host Object Support – parent global Object	High	[ESMP_Suite] The TCs from section 6.1
Host Object Support – navigator Object	High	[ESMP_Suite] The TCs from section 9.2
Host Object Support – history Object	High	[ESMP_Suite] The TCs from section 9.3
Host Object Support – location Object	High	[ESMP_Suite] The TCs from section 9.4

Table 5 User Agent ETR for ESMP Priorities for IOP Test

B.1.2.1 Optionality

REQUIREMENT	DESCRIPTION	TEST TOOL

The test campaign needs to verify the User Agent to support the mandatory subset of ESMP, allowing any optionally defined additions. (optionality)	The major optional chunk is DOM Structural mutationW3C DOM. Any testing associated with this should be separable from other testing material.	[ESMP_Suite] TCs form chapter 11 related with SM (8 TCs)

Table 6 User Agent Optionality Requirements

B.1.3 Proxy/Gateway/PEP

B.1.3.1 Deployment

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the proxy/gateway/PEP to can correctly transmit content type XHTML documents with embedded ECMAScript to user agents that support the content type	Content that is acceptable to a client must not be transformed or modified in any way. HTTP accept headers should have been tested when testing XHTML-MP, but those tests should also account for embedded script	[ESMP_Suite] All the TCs from chapter 7 with the usage of the Proxy/Gateway/PEP
Any Proxy/Gateway or PEP MUST be able to support both UTF-8 and UTF-16 encodings	The W3C specification assumes UTF-8 for XHTML, but the ECMAScript assumption is for UTF-16. The OMA spec allows either to be served and the client may choose. This is because UTF-16 is the common basis for Chinese/Japanese/Korean implementations, and UTF-8 is the common basis for Latin implementations	[ESMP_Suite] The TCs from section 5.2 with the usage of the Proxy/Gateway/PEP

Table 7 ESMP Gateway Deployment Requirements

B.1.3.2 Interoperability

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify that the proxy/gateway/PEP will react identically to all other gateways in error situations	Posting of errors to the client must occur using standard HTTP error codes and the codes used must be consistent from proxy to proxy	[ESMP_Suite] The TCs from section 6.9 with the usage of the Proxy/Gateway/PEP

Table 8 ESMP Gateway Semantic Interoperability Requirements

B.1.3.3 Optionality

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the proxy/gateway/PEP to will transform content to a useable content type encoding, if capable of	Character encoding mismatches between content server and client should be handled by an intervening proxy that advertises the capability to transform character sets.	[ESMP_Suite] The TCs 49.01.01/02 from section 5.2 with the usage of the Proxy/Gateway/PEP

doing so.		

Table 9 ESMP Gateway Optionality Requirements

B.1.3.4 Content Server

REQUIREMENT	DESCRIPTION	TEST TOOL
The test campaign needs to verify the content server to will respond correctly to URL requests	While not explicitly an ESMP requirement, the content server must respond correctly to URL requests for script content.	New test cases have to be created
The test campaign needs to verify the content server to will serve appropriate content type if available	Content, if not available must generate the appropriate HTTP errors.	New test cases have to be created
The test campaign needs to verify the content server to will react identically to all other content servers in error situations.	Posting of errors to the client must occur using standard HTTP error codes and the codes used must be consistent from content server to content server.	New test cases have to be created

Table 10 ESMP Content Server Requirements

B.1.3.5 Content Generation

REQUIREMENT	TEST TOOL
The test campaign needs to verify the tools used to generate test content should be able to generate content that is restricted to the allowed language constructs and objects defined in ESMP.	Since ESMP is a super set of ECMA262, DOM2HTML and DOM2CORE the content can be validated using wired world clients. As stated on [ESMP], Internet Explorer and Netscape are the reference browsers and can be used to validate all the scripts.
The test campaign needs to verify the tools should be able to validate script content based on the definition of ESMP	Since ESMP is a super set of ECMA262, DOM2HTML and DOM2CORE the content can be validated using wired world clients. As stated on [ESMP], Internet Explorer and Netscape are the reference browsers and can be used to validate all the scripts.

Table 11 ESMP Content Generation Requirements

B.2 WAE

This Browsing 2.2 ETP chapter will only cover the basic WAE requirements. Specific other Browsing 2.2 related requirements will be processed inside other ETP chapters (XHTML-MP, WML, WCSS, ESMP, WTA, Pictogram, Cashing Model) while specific requirements will be processed inside different future planned ETP's (Persistent Storage and EFI).

The WAE (Media-Types) specifications related Browsing 2.2 ETR requirements can partly be matched towards the existing WAP 2.0 testsuite, [VSWAP 2.0].

B.2.1 User Agent

REQUIREMENT	TEST TOOL
The test campaign needs to ensure the content types required to be supported are verified. See Table 6.1 within [WAE] section 6.1	[VSWAP 2.0] test case "wae2/media/types/1" Extra test case(s) needed for Support for MIME Media Types, using information obtained from the acceptheader.
The test campaign must verify the User Agent support images when it support the images. Support for grahical images is optional but where a UA supports graphical images the UA needs to support WBMP. See [WAE] section 6.6.	[VSWAP 2.0] test case "wae2/media/images/gcrules/1".
The test campaign must verify exchange via the supported session protocol(s) and/or WDP and that the minimum properties of "Name" and "Telephone Number" are supported for display and, where supported, transmission when claimed to be supported. See [WAE] section 6.7. (Support for vCard [VCARD] is optional. A UA MAY support vCard via the supported session protocol(s), WSP and/or W-HTTP, and/or WDP exchange).	[VSWAP 2.0] test cases "wae2/media/vcard/cardrules/1, /3 and /4".
The test campaign must verify that the User Agent verify exchange via the supported session protocol(s) and /or WDP and the requirement to display the vEvent object when claimed to be supported. See [WAE] section 6.8. (Support for vCalendar [VCAL] is optional. A UA MAY support vCalendar via the supported session protocol(s), WSP and/or W-HTTP, and/or WDP exchange).	[VSWAP 2.0] test cases "wae2/media/vcal/calrules/1 and/3".
The test campaign needs to verify the UA supported multipart message encoding formats. Multipart/mixed and multipart/related and multipart/alternative are all optional though multipart/mixed and multipart/related are more strongly recommended. Multipart/form-data is required and needs to be tested. See also [WAE] section 6.9.	[VSWAP 2.0] test cases "wae2/media/multipart/form/1, /3 and /4".
The test campaign must verify support for the supported hypermedia transfer service. The test campaign should be able to verify the hypermedia tranfer service through the execution of tests defined in this ETR between the UA and content server with and/or without the proxy/gateway/PEP in between depending on the device and content server features. UAs are required to support the hypermedia transfer service, i.e. [WSP] and/or [W-HTTP]. See also [WAE] section 7.1.	No test cases available.

The test campaign needs to verify the minimum URI length. (1024 octets) See also [WAE] section 7.2.	[VSWAP 2.0] test case "wae2/features/uri/1"
The test campaign needs to verify retrieving a resource specified by a HTTP URI scheme is communicated to the proxy/gateway/PEP or content server, depending on deployment topology being tested, using either [W-HTTP] or [WSP]. See also [WAE] section 7.2.	Untestable.
The test campaign needs to verify UAs support the HTTPS URI scheme and that when retrieving a resource using the HTTPS URI scheme the behavior is consistent with that specified in [WAE] section 7.2.2, i.e. the establishment of a secure connection across all links between UA and content server. The use of https: implies that there is a secure linkage end-to-end. [WAE] specifies that if https: is used between a client and a gateway, the system must insure that similar security constraints are applied between the gateway and the server. See also [WAE] section 7.2.2.	No test cases available.
The test campaign needs to verify the UA reporting an error to the user when the secure connection cannot be established. See also [WAE] section 7.2.2.	Untestable.
The test campaign need to verify that the User Agent support other schemes. See also [WAE] section 7.2.3.	No test cases available. Since the other schemes are related to functionalities covered in different ETP (like for example EFI, Persistent Storage, Pictograms etc) they do not need testing within the scope of the standard WAE test cases.
The test campaign needs to verify UA support for the character sets defined in [WAE] section 7.6.1, i.e. text encoded in UTF-8 and UTF-16. The test campaign needs to verify the treatment of character encoding and the reporting of errors to the user when the document includes unknown characters.	[VSWAP 2.0] test cases "wae2/features/internationalisation/encoding/1 till /5"
The test campaign needs to verify the UA supporting the advertising of its characteristics does so using HTTP accept headers regardless of any [UAPROF] support. See also [WAE] section 7.7.	No test cases available.
The test campaign should verify the UA supports for the advertised characteristics and needs to verify support for all advertised characteristics specified by OMA. See also [WAE] section 7.7.	No test cases available.
The test campaign needs to verify the navigation history model as defined in [WAE] section 7.13.1	[VSWAP 2.0] test cases "wae2/features/uabehaviour/navhistory/rules/1,/3, /4"
The test campaign needs to verify the availability of the BACK key at all times as defined in [WAE] 7.13.2. and the test campaign needs to verify the resulting behaviour from the activation of the BACK key as defined in [WAE] 7.13.2.	[VSWAP 2.0] test cases "wae2/features/uabehaviour/backkey/rules/1, /3, /4 and /5"

Table 12 WAE UA Deployment Requirements

REQUIREMENT	PRIORITY	TEST TOOL
Verify UA declares acceptance of all supported content types (those from [WAE] section 6.1 plus common media types, with priority on the following content types: For presentation in the browser: WML textual form, WML1 binary form, XHTML Basic, XHTML Mobile Profile, WAP CSS, WAP BMP, BMP, GIF, JPEG, PNG For download: AMR, MIDI, MP3, WAV, H.263 Video (3GPP) Verify UA Support (via in-line/direct presentation or other supported handling) for all declared content types.	High	[VSWAP 2.0] test case "wae2/media/types/1" Extra test case(s) needed for Support for MIME Media Types, using information obtained from the acceptheader. However some of them may be tested via test cases found in the other ETP chapters (e.g. XHTML, WCSS.etc.) For Download content types specific test cases need to be developed (see Download ETS).
Verify session establishment via supported WAP1 protocols, including secure and non-secure session protocols	High	Explicitly tested using test cases within ETP chapter for WML. However some specific general tests could be developed or chosen from the VSWAP 2.0 suite covering normal and secure (WTLS) WAP sessions.
Verify data exchange via all supported protocols, including non-secure and secure transport protocols of WAP1 and WAP2	High	Probably un-testable or protocol testing is being considered as part of the test campaign.
Verify supported hypermedia transfer services, including the GET, HEAD, and POST methods, and status codes 200, 302	High	Some specific tests could be developed or chosen from the VSWAP 2.0 suite.
If supported, verify support for the HTTP Refresh header.	High	HTTP refresh header is not mentioned inside the WAE specifications. There is no test case available yet and if these have to be tested, then test cases have to be developed.
Verify minimum URI length support in all hypermedia transfer and markup language features that use URI's, with priority on: Anchor tags, Embedded object references, HTTP 302 redirect (Location header), WML-specific features, Push Service Indication and Service Loading URI's	High	Very likely un-testable.
Verify supported URI schemes, including "http", "https", "mailto", and "wtai"	High	Some specific test cases should be developed.
Verify operation of softkeys when defined in applications. [WML]	High	Explicitly tested using test cases within ETP chapter for WML.
Verify the back key at all times	High	[VSWAP 2.0] test cases "wae2/features/uabehaviour/backkey/rules/1, /3, /4 and /5"

Table 13 WAE Use Agent Interoperability requirements

B.2.1.1 Proxy/Gateway/PEP

REQUIREMENT	TEST TOOL
The test campaign needs to verify the proxy informing the UA of an error in compilation using HTTP status code 502 “Bad Gateway”.	Some specific test cases should be developed.
The test campaign should verify transparency in support for images.	Some specific test cases should be developed.
The test campaign should verify transparency in support for vCard.	Some specific test cases should be developed.
The test campaign should verify transparency in support for vCalendar.	Some specific test cases should be developed.
The test campaign should verify transparency in support for multipart.	Some specific test cases should be developed.
The test campaign needs to verify the proxy/gateway/PEP conversion from standard Internet multipart to WAP specific multipart where devices use WSP to connect to the proxy/gateway/PEP.	[VSWAP 2.0] test case “wae2/media/multipart/form/4” can be used as basic test case but more specific testing should be considered as “un-testable”.
The test campaign needs to verify the support of HTTP/1.1 and/or WSP as the Hypermedia Transfer Service.	No test cases available.
The test campaign needs to verify proxy/gateway/PEPs support for WSP cached request headers.	No test cases available.
The test campaign needs to verify support for the minimum URI length. (1024 octets) The test campaign needs to verify retrieving a resource specified by a HTTP URI scheme is communicated to the proxy/gateway/PEP or content server, depending on deployment topology being tested, using either [W-HTTP] or [WSP]. The test campaign needs to verify support for the HTTPS URI scheme and that when retrieving a resource using the HTTPS URI scheme the behaviour is consistent with that specified in [WAE] section 7.2.2, i.e. the establishment of a secure connection across all links between UA and content server. The test campaign needs to verify the Proxy reports an error to the user when the secure connection cannot be established. [WAE] section 7.1.4	[VSWAP 2.0] test case “wae2/features/uri/2” can be used as basic test case but more specific testing should be considered as “un-testable”.
The test campaign needs to verify transformation of character encoding when the UA does not support the original character encoding in internationalisation. [WAE] section 7.6.1	No test cases available. These have to be developed
The test campaign needs to verify proxy/gateway/PEP support for the character sets defined in [WAESpec] section 7.6.1, i.e. text encoded in UTF-8 and UTF-16.	[VSWAP 2.0] test case “wae2/features/internationalisation/encoding/3 and/4”
The test campaign needs to verify treatment of character encoding and reporting of errors to the user when the document includes unknown characters.	[VSWAP 2.0] test case “wae2/features/internationalisation/encoding/5”

[WAE] section 7.6.1	
The test campaign needs to verify the assumed values of HTTP/1.1 headers as defined in [WAE] section 7.7.1.	Test case need to be developed for this.
The test campaign needs to verify any claimed honouring of device preferences as defined in [WAESpec] section 7.7.1	Test case need to be developed for this.

Table 14 WAE Gateway Deployment requirements

B.2.1.2 Priorities

REQUIREMENT	PRIORITY	Test Tool
Verify session establishment via supported WAP1 protocols, including secure and nonsecure session protocols	High	Explicitly tested using test cases within ETP chapter for WML. However some specific general tests could be developed or chosen from the VSWAP 2.0 suite covering normal and secure (WTLS) WAP sessions.
Verify data exchange via all supported protocols, including nonsecure and secure transport protocols of WAP1 and WAP2	High	Probably untestable or protocol testing is being considered as part of the testcampaign.
Verify supported hypermedia transfer services, including the GET, HEAD, and POST methods, and status codes 200, 302	High	Some specific tests could be developed or chosen from the VSWAP 2.0 suite.
Verify caching/forwarding of session headers (WAP1) and request headers (WAP1 and WAP2)	High	Specific test cases should be developed to test this.
Verify proxied request accept header includes all content types declared by the UA, and additionally content types supported for translation to UA-accepted content types.	High	Test case should be developed for this. (unless it is explicitly tested by UA test cases)
Verify delivery of UA-declared content types (those from [WAE] section 6.1 plus common media types, with priority on the following content types: WML textual form, WML1 binary form, XHTML Basic, XHTML Mobile Profile, WAP CSS, WAP BMP, BMP, GIF, JPEG, PNG, AMR, MIDI, MP3, WAV, H.263 Video (3GPP)	High	[VSWAP 2.0] test case “wae2/media/types/1” Extra test case(s) needed for Support for MIME Media Types, using information obtained from the acceptheader. However some of them may be tested via test cases found in the other ETP chapters (e.g. XHTML, WCSS.etc.) For Download content types specific test cases need to be developed (see Download ETS).
Verify nonsecure and secure delivery of content types as above via proxy/Gateway/PEP and directly to content server	High	Explicitly tested using test cases within ETP chapter for WML. However some specific general tests could be developed or choosen from the VSWAP 2.0 suite covering normal and secure (WTLS) WAP sessions.

Table 15 WAE Proxy/Gateway/PEP ETR Priorities for IOP Test

B.2.1.3 Content Server

REQUIREMENT	TEST TOOL
HTTP server – with or without Proxy/GW/PEP The test campaign needs to ensure all the content type required to be supported for the Browsing V2.3 feature support are verified as supported by the Content Server, for the Media requirements defined in [WAE] section 6.1. (Section 6.1 defines the minimum set of media types a UA needs to support for the features supported.)	Test case should be developed for this. (unless it is explicitly tested by UA test cases)
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify support as defined in [WAE] section 6.7 when vCard is supported.	[VSWAP 2.0] test cases “wae2/media/vcard/cardrules/1, /3 and /4”.
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify support as defined in [WAE] section 6.8 when vCalendar is supported.	[VSWAP 2.0] test cases “wae2/media/vcal/calrules/1 and /3”.
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify claimed support for multipart as defined in [WAE] section 6.9.	[VSWAP 2.0] test cases “wae2/media/multipart/form/1”.
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify the support of HTTP/1.1 for the request and response of resources specified by URLs on the content server.	Test case should be developed for this
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify the Support for the Client Header Handling should be catered for via the “support for HTTP/1.1 and WSP” section above or the support for Client Header Handling for Proxy/Gateway/PEP (section 6.2.2) or the HTTP/1.1/WSP Server ETRs in section 6.2.3.1.2.	Test case should be developed for this
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify support for the minimum URI length. While this requirement in [WAE] specifically calls out UAs and Proxies it is equally applicable to Servers since the URIs would be sent by the UA directly or via Proxy to the content server.	[VSWAP 2.0] test case “wae2/features/uri/2”.
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify retrieving a resourse specified by a HTTP URI scheme. The request and response may be made using [WSP] or [W-HTTP] if via a proxy/gateway/PEP or direct to the content server, or via HTTP if via a proxy/gateway/PEP.	Test case should be developed for this.
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify support for the HTTPS URI	Test case should be developed for this.

scheme and that when retrieving a resource using the HTTPS URI scheme the behaviour is consistent with that specified in [WAE] section 7.2.2, i.e. the establishment of a secure connection across all links between UA and content server.	
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify support for the character sets defined in [WAE] section 7.6.1, i.e. text encoded in UTF-8 and UTF-16.	[VSWAP 2.0] test cases "wae2/internationalisation/encoding/3 and /4".
HTTP server – with or without Proxy/GW/PEP Verify support for HTTP/WSP accept headers. [WAE] section 7.7.1.	Test case should be developed for this
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify behavior when the proxy content server sends a response acceptable to the UA.	Test case should be developed for this
HTTP server – with or without Proxy/GW/PEP The test campaign needs to verify any claimed honoring of device preferences as defined in [WAE] section 7.7.1	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to ensure all the content type required to be supported by UAs are verified as supported by the Proxy/Gateway/PEP functionality within the content server for support the Media requirements defined in [WAE] section 6.1. (Section 6.1 defines the minimum set of media types a UA needs to support for the features supported.)	Test case should be developed for this. (unless it is explicitly tested by UA test cases)
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify function of the content server informing the UA of an error in compilation using HTTP status code 502 "Bad Gateway".	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign should verify transparency for support for images. (No specific ETRs other than transparency (section 6.1). The content server support for images is defined in section 6.2.3.1.1.)	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify support as defined in [WAE] section 6.7 when vCard is supported.	Could be basically tested using [VSWAP 2.0] test cases "wae2/media/vcard/cardrules/1, /3 and /4".
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify support as defined in [WAE] section 6.8 when vCalendar is supported.	Could be basically tested using [VSWAP 2.0] test cases "wae2/media/vcal/calrules/1 and /3".
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify the use of the WAP specific	New test cases should be developed. Basically [VSWAP 2.0] test case

multipart where [WSP] is used standard multipart where [W-HTTP] is used. It is immaterial if the content server translates the multipart or creates the variants directly in this deployment.	"wae2/media/multipart/form/4" could be used in addition.
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify direct support for HTTP/1.1 and WSP by servers that are access with a proxy/gateway or PEP. (The test campaign needs to verify the support of HTTP/1.1 and/or WSP as the Hypermedia Transfer Service.)	New test cases should be developed for this.
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify support for the minimum URI length.	[VSWAP 2.0] test case "wae2/features/uri/2".
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify retrieving a resource specified by a HTTP URI scheme is communicated to the content server using either [W-HTTP] or [WSP].	Test case should be developed for this.
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify support for the HTTPS URI scheme and that when retrieving a resource using the HTTPS URI scheme the behavior is consistent with that specified in [WAE] section 7.2.2, i.e. the establishment of a secure connection across all links between UA and content server.	Test case should be developed for this.
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify the content server reports an error to the user when the secure connection cannot be established.	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify transformation of character encoding when the UA does not support the original character encoding for Internationalization.	No test cases available. These have to be developed
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify support for the character sets defined in [WAESpec] section 7.6.1, i.e. text encoded in UTF-8 and UTF-16.	[VSWAP 2.0] test cases "wae2/internationalisation/encoding/3 and /4".
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify treatment of character encoding and reporting of errors to the user when the document includes unknown characters.	[VSWAP 2.0] test case "wae2/internationalisation/encoding/5".
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP Verify support for HTTP/WSP accept headers.	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP	Test case should be developed for this

The test campaign needs to verify the assumed values of HTTP/1.1 headers as defined in [WAESpec] section 7.7.1.	
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify behavior when the content server cannot send a response acceptable to the UA.	Test case should be developed for this
HTTP/1.1/WSP Server – no Proxy/Gateway/PEP The test campaign needs to verify any claimed honoring of device preferences as defined in [WAESpec] section 7.7.1	Test case should be developed for this

Table 16 WAE Content Server Deployment requirements

B.2.1.4 Content Server

REQUIREMENT	PRIORITY	TEST TOOL
Verify supported hypermedia transfer services, including the GET, HEAD, and POST methods, and status codes 200, 302	High	Some specific tests could be developed or chosen from the VSWAP 2.0 suite.
Verify delivery of UA-declared content types supported by the content server (those from [WAE] section 6.1 plus common media types, with priority on the following content types: WML textual form, WML1 binary form, XHTML Basic, XHTML Mobile Profile, WAP-CSS, WAP-BMP, BMP, GIF, JPEG, PNG, AMR, MIDI, MP3, WAV, H.263 Video (3GPP)	High	[VSWAP 2.0] test case “wae2/media/types/1” Extra test case(s) needed for Support for MIME Media Types, using information obtained from the acceptheader. However some of them may be tested via test cases found in the other ETP chapters (e.g. XHTML, WCSS.etc.) For Download content types specific test cases need to be developed (see Download ETS).
Verify nonsecure and secure delivery of content types as above via proxy/Gateway/PEP and directly to content server (Note: for the sake of clarity, the specification assumes WSP server to logically be combination of Proxy and content server)	High	Explicitely tested using test cases within ETP chapter for WML. However some specific general tests could be developed or choosen from the VSWAP 2.0 suite covering normal and secure (WTLS) WAP sessions.

Table 17 WAE Content Server Interoperability requirements

B.3 Test Tools WAE Media Types

The WAE Media Type related Browsing 2.2 ETR requirements can be matched to the WAE test suite [VSWAP 2.0] assertions in the following manner.

B.3.1 User Agent

B.3.1.1 Media Types

REQUIREMENT	TEST TOOL
The test campaign needs to verify that when a device claims to support images it advertised support for WBMP.	[VSWAP 2.0].
The test campaign needs to verify that when a device claims to support images it accepts and renders them according to the definition in [WAEMedia].	[VSWAP 2.0].

Table 18 WAE Media Type, User Agent Deployment requirements

B.3.1.2 Optionality

REQUIREMENT	TEST TOOL
The test campaign should verify animated WBMP is the device claims to support it	[VSWAP 2.0].

Table 19 WAE Media Type, User Agent Optionality

B.3.1.3 Proxy/Gateway/PEP

REQUIREMENT	TEST TOOL
the [WAE] tests for proxy transparency for requests, responses, passing on accept headers etc. need to be verified for WBMP.	[VSWAP 2.0].

Table 20 WAE Media Type Gateway Deployment requirements

B.4 WCSS

The Browsing 2.2 ETR requirements can be matched to the WCSS test suite [WCSS_Suite] assertions in the following manner.

REQUIREMENT	TEST TOOL
Syntax and parsing. In particular it MUST be tested that the UA ignores properties it does not support, and invalid syntax.	[WCSS_Suite] Chapter 5.
The test campaign MUST test that the UA is able to cascade style rules from external style sheets, the "style" attribute, and the <style> element.	[WCSS_Suite] Most of the test cases test external style sheets.
The test campaign MUST test with XHTML mobile profile documents. Also, it must be tested that style sheets can be inside the XHTML <style> elements and the "style" attribute.	[WCSS_Suite]. All the test cases are XHTML documents. Internal style sheets are tested e.g. in chapter 9.7.
For external style sheets the test campaign MUST test that the document with the "text/css" MIME media types are handled as WCSS style sheets.	[WCSS_Suite] Most of the test cases test this.

Table 21 WCSS General Requirements

<table>
<tr><th>REQUIREMENT</th><th>PRIORITY</th><th>TEST TOOL</th></tr>
<tr><td>Verify WCSS V1.1 mandatory features, with priority on:
external stylesheets
internal stylesheets
style attribute
extensions: input
Pseudo-classes</td><td>High</td><td>All WCSS V1.1 mandatory SCRs are covered by [WCSS_Suite]. In particular:
<table>
<tr><td>external style sheets</td><td>Most of the test cases.</td></tr>
<tr><td>internal style sheets</td><td>E.g. Chapter 9.7</td></tr>
<tr><td>style attribute</td><td>Chapter 4.1 (45.14.12)</td></tr>
<tr><td>extensions: input</td><td>Chapter 18</td></tr>
<tr><td>Pseudo-classes</td><td>Chapter 4.1 (45.14.05 and 45.14.06)</td></tr>
</table></td></tr>
<tr><td>Verify WCSS V1.1 optional features with priority on:
extensions: -wap-accesskey</td><td>High</td><td>All WCSS V1.1 optional SCRs are covered by [WCSS_Suite]. In particular:
<table>
<tr><td>extensions: -wap-accesskey</td><td>Chapter 17.</td></tr>
</table></td></tr>
<tr><td>Box model properties (e.g. margin, border)</td><td>High</td><td>[WCSS_Suite] Chapter 9.</td></tr>
<tr><td>Lists</td><td>High</td><td>[WCSS_Suite] Chapter 12.</td></tr>
<tr><td>Text indentation</td><td>High</td><td>[WCSS_Suite] Chapter 13.1</td></tr>
<tr><td>White space</td><td>High</td><td>[WCSS_Suite] Chapter 13.5</td></tr>
<tr><td>Float Positioning</td><td>High</td><td>[WCSS_Suite] Chapter 15.2</td></tr>
<tr><td>CSS Extension: Marquee</td><td>High</td><td>[WCSS_Suite] Chapter 16.</td></tr>
</table>

Table 22 WCSS Prioritized requirements

B.5 WML

ACTOR	TYPE OF REQUIREMENT	COVERED BY	REQUIREMENT	TEST TOOL
User Agent	Deployment	WML 1.3	The test campaign for deployment of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	-
User Agent	Interoperability	WML 1.3	The test campaign for interoperability of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	[VSWAP 2.0]
User Agent	Optionality	WML 1.3	The test campaign for optionality of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	[VSWAP 2.0]
Proxy/Gateway/PEP	Deployment	WML 1.3	The test campaign for deployment of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	-
Proxy/Gateway/PEP	Interoperability	WML 1.3	The test campaign for deployment of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	[VSWAP 2.0]
Proxy/Gateway/PEP	Optionality	WML 1.3	The test campaign for deployment of WML 1.3 should use the test assertions and expected responses of the WAP certification programme bearing in mind any outcomes of the PRs raised.	[VSWAP 2.0]
Content Server ETR	Deployment	N/A	-	
Content Server ETR	Interoperability	N/A	-	
User Agent	Deployment	WML 2.0	The test campaign needs to verify the User Agent can handle WML2, XHTML Basic, and XHTML MP content types	[XHTML_Suites] Overlap with XHTML & [VSWAP 2.0]
User Agent	Deployment	WML 2.0	The test campaign MUST verify that the User Agent can advertise the content type properly to process the WML1.3 document on the content server.	-
User Agent	Interoperability	WML 2.0	The test campaign needs to verify that	[VSWAP 2.0]

			the User Agent correctly handles Back key behaviour.	
User Agent	Interoperability	WML 2.0	The test campaign needs to verify the User Agent shows WML 1.3 compatible behavior when it processes documents from WML1.3 content server.	[VSWAP 2.0]
User Agent	Optionality	WML 2.0	-	-
Proxy/Gateway/PEP		WML 2.0	The test campaign needs to verify the Proxy/Gateway/PEP to process the Accept-Header from the user agent to produce WML2 document from WML1.3 documents in the content server.	TBD
Proxy/Gateway/PEP		WML 2.0	The test campaign needs to verify the Proxy/Gateway/PEP to convert WML1.3 documents to WML2 document when the user agent request only WML2 document excluding WML1.3 content type.	TBD
User Agent	Deployment	WML 2.0	The test campaign should put a low priority with the no proxy/Gateway/PEP deployment case.	-
User Agent	Interoperability	WMLScript	The test campaign must be aligned to the past WAP test suites.	[VSWAP 2.0]
Proxy/Gateway/PEP		WMLScript	The test campaign should use the past WAP test suites.	[VSWAP 2.0]
Proxy/Gateway/PEP		WMLScript	The test campaign should use the past WAP test suites.	[VSWAP 2.0]
User Agent	Deployment	WMLScript Libraries	The test campaign needs to verify that User Agents that support WMLScript support the mandatory WMLScript Libraries.	-
User Agent	Interoperability	WMLScript Libraries	The test campaign should be aligned to the past WAP test suites.	[VSWAP 2.0]

Table 23 WML Requirements

B.6 XHTML

ACTOR	TYPE	REQUIREMENT	TEST TOOL
User Agent (Client)	Deployment	The test campaign needs to verify the User Agent can handle XHTML Basic and XHTML MP content types.	[XHTML_Suites]
		The test campaign needs to verify the User Agent can support deployment directly to the content server when they support it.	[XHTML_Suites]
		The test campaign needs to verify the User Agent is able to process the document type definition specified as supported.	[XHTML_Suites]
		The test campaign needs to verify the User Agent conforms to the user agent conformance specified in [XHTMLMP] Section 7.2	[XHTML_Suites]
	Interoperability	The test campaign needs to verify that all User Interface Widget implementations are consistent with the semantic intent of [XHTMLMP], and with a representative sample of other client implementations.	[XHTML_Suites]
	Optionality	The test campaign needs to verify the User Agent to support WCSS when it supports Style Sheets.	[XHTML_Suites]
Proxy/Gateway/ PEP	Optionality	The use of Proxy/Gateway/PEP is optional and a low priority in the test campaign.	[XHTML_Suites]
Content Server	Deployment	The test campaign needs to verify the User Agent to support deployment directly to the user agent.	[XHTML_Suites]
		XHTMLMP V 1.1 Specific	
User Agent (Client)	Deployment	The test campaign needs to verify the User Agent to support deployment directly to the content server.	
		The test campaign needs to verify the User Agent to implement ESMP.	ESMP ETP
		Test campaign needs to support both of XHTMLMP processing and ESMP processing.	ESMP ETP
		The test campaign needs to verify the User Agent to process the elements specified in [XHTMLMP10] Section 9.	
		The test campaign needs to verify the User Agent to process and execute the script semantics in [XHTMLMP10] Section 9.2.	
		The test campaign needs to verify the User Agent to process the events and event handlers specified in [XHTMLMP10] Section 10.2	

		The test campaign needs to verify the User Agent to process the event handler registration specified in [XHTMLMP10] Section 10.4.	
	Interoperability	The test campaign needs to verify that all User Interface Widget implementations are consistent with the semantic intent of [XHTMLMP], and with a representative sample of other client implementations.	
Proxy/GW/PEP		The use of Proxy/Gateway/PEP is optional and test campaign puts a low priority.	
Content Server	Deployment	The test campaign needs to verify the content to support deployment directly to the user agent.	

Table 24 XHTML Requirements