URI Schemes for the Mobile Applications Environment

Approved Version 1.0 – 26 Jun 2008

Open Mobile Alliance

OMA-TS-URI_Schemes-V1_0-20080626-A

Contents

Figures

Tables

1. Scope

This document describes support for a set of URI Schemes that are to be ubiquitous in the mobile environment. By ubiquitous, we mean these schemes may appear and apply anywhere within the set of applications, services and enablers that are the mobile environment. The document specifies a set of technical requirements for these schemes. The intention is to create a high level of interoperability within the mobile industry when implementing these schemes, and to insure that this set of URI schemes is available wherever useful. In those cases where the source standardization may be ambiguous or incompletely defined for the mobile environment, this document may add or clarify technical specifications in the source standard(s).

The document does not limit the use of other URI schemes within the mobile application environment.

The scope of this specification encompasses all user agents that operate within the OMA Mobile Applications Environment.

2. References

2.1 Normative References

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/

[OMAMMSENC] "OMA Multimedia Messaging Service Encapsulation Protocol", Version 1.2, Open Mobile Alliance™, OMA-MMS-ENC-V1_2-20050301-A, URL:http//www.openmobilealliance.org/

[OMATSPOCCP] "OMA POC Control Plane", Version 1.0, Open Mobile Alliance™, OMA-TS-PoC-ControlPlane-V1_0-20060127-C, URL:http//www.openmobilealliance.org/

[OMATSSIMPLE] "OMA Technical Specification – IM SIMPLE", Draft Version 1.0,, Open Mobile Alliance™, OMA-TS-IM-SIMPLE-V1_0_0-20060223-D, URL:http//www.openmobilealliance.org/

[OMAURISREQ] "URI Schemes Requirements", Version 1.0, Open Mobile Alliance™, OMA-RD_URI-Schemes-V1_0, URL:http//www.openmobilealliance.org/

[RFC2047] "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", K. Moore, November 1996, URL: http://www.ietf.org/rfc/rfc2047.txt

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

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

[RFC2368] " The mailto URL Scheme", P. Hoffman, L. Masinter, J. Zawinski, July 1998, URL: http://www.ietf.org/rfc/rfc2368.txt

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, June 1999, URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC2818] "HTTP over TLS", E. Rescorla, May 2000, URL: http://www.ietf.org/rfc/rfc2818.txt

[RFC2822] "Internet Message Format", P. Resnick, April 2001, URL: http://www.ietf.org/rfc/rfc2822.txt

[RFC3261] "SIP: Session Initiation Protocol", J. Rosenberg, H. Schulzrinne, G.Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, June 2002, URL: http://www.ietf.org/rfc/rfc3261.txt

[RFC3601] "Text String Notation for Dial Sequences and Global Switched Telephone Network (GSTN) / E.164 Addresses", C. Allocchio, September 2003, URL: http://www.ietf.org/rfc/rfc3601.txt

[RFC3840] "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", J. Rosenberg, H. Schulzrinne, P. Kyzivat, August 2004, URL: http://www.ietf.org/rfc/rfc3840.txt

[RFC3966] "The tel URI for Telephone Numbers", H. Schulzrinne, December 2004, URL: http://www.faqs.org/ftp/rfc/rfc3966.txt

[RFC3986] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, January 2005, URL: http://www.ietf.org/rfc/rfc3986.txt

[SMSSCHEME] "URI scheme for GSM Short Message Service", Internet Draft, E. Wilde, A. Vaha-Sipila, July 9,2007, (Expires January 10, 2008), URL: http://www.ietf.org/internet-drafts/draft-wilde-sms-uri-13.txt (altURL: http://dret.net/netdret/docs/draft-wilde-sms-uri-13.txt)

[WAESPEC] "Wireless Application Environment Specification", Candidate Version 2.3, Open Mobile Alliance™, February 27, 2007, URL:http://www.openmobilealliance.org/ftp/Public_documents/BT/MAE/Permanent_documents/OMA-WAP-TS_WAESpec-V2_3-20070227-C.zip

2.2 Informative References

[RFC4395] "Guidelines and Registration Procedures for new URI Schemes", T.Hansen, T. Hardie, L. Masinter, February 2006, URL: http http://www.ietf.org/rfc/rfc4395.txt

[RFC3305] "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations" , M. Mealling, R. Denenberg, August 2002, URL: http://www.ietf.org/rfc/rfc3305.txt

[W3CWEBARCH] "Architecture of the World Wide Web, Volume One", W3C Recommendation, Ian Jacobs, Norman Walsh, December 15, 2004, URL: http://www.w3.org/TR/webarch/

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” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

Syntax definitions are given using the Augmented BNF for Syntax Specifications [RFC2234].

3.2 Definitions

Percent-encoding	A mechanism that is used to represent a data octet when that octet's corresponding character is outside the allowed set or conflicts with a reserved character's purpose as a delimiter. A percent-encoded octet is encoded as a character triplet, consisting of the percent character "%" followed by the two hexadecimal digits representing that octet's numeric value. See [RFC3986]], section 2.
URL-encoding	A commonly used term for percent-encoding.
URI Scheme	The “scheme” part of a Uniform Resource Identifier as specified in [RFC3986]

3.3 Abbreviations

IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
MAE	Mobile Applications Environment
OMA	Open Mobile Alliance
RFC	Request For Comments – the publication method used by IETF
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name

4. Introduction (Informative)

The OMA Mobile Applications Environment (MAE) has graduated to become a first class application and web environment. Applications such as web browsers, messaging applications, email applications and others have joined the traditional telephone call as primary uses for the ubiquitous mobile device. Along with the explosion of applications comes a call for the ability to integrate these applications in a standardized manner. Standardized integration mechanisms give the development and content community well known ways of coordinating various applications and allowing the sharing of function and content to create higher-level services.

The public Internet endorses a standardized syntax for the identification of resources [RFC3986], such as documents, applications, etc. known as URIs. Included in the general identification of a resource is specific information for resolving the the URI scheme, and optionally parameters for carrying content and context specific information. While there have been hundreds of schemes proposed and used in the internet, only a small number (~75) have been registered and standardized by the IETF. A number of schemes are applicable to the MAE and the user agents associated with a mobile device. These "URI schemes", along with the content of the URI itself, if handled in a device independent, interoperable manner represent a standardized mechanism for sharing internet resources and allowing user agent services to invoke other user agent services [OMAURISREQ].

As an example, the URI scheme mailto: is well known to people familiar with Internet services. When applied as the scheme part of a URI, it says that the rest of the URI is to interpreted and processed within the context of an email subsystem, according to a standardized set of mechanisms [RFC2368]. In practice this means that an operating environment turns the URI over to an application that can process email requests. Thus, with a standard set of agreed URI schemes, we have a way for content developers to share resources and orchestrate the integrated use of multiple logical user agents.

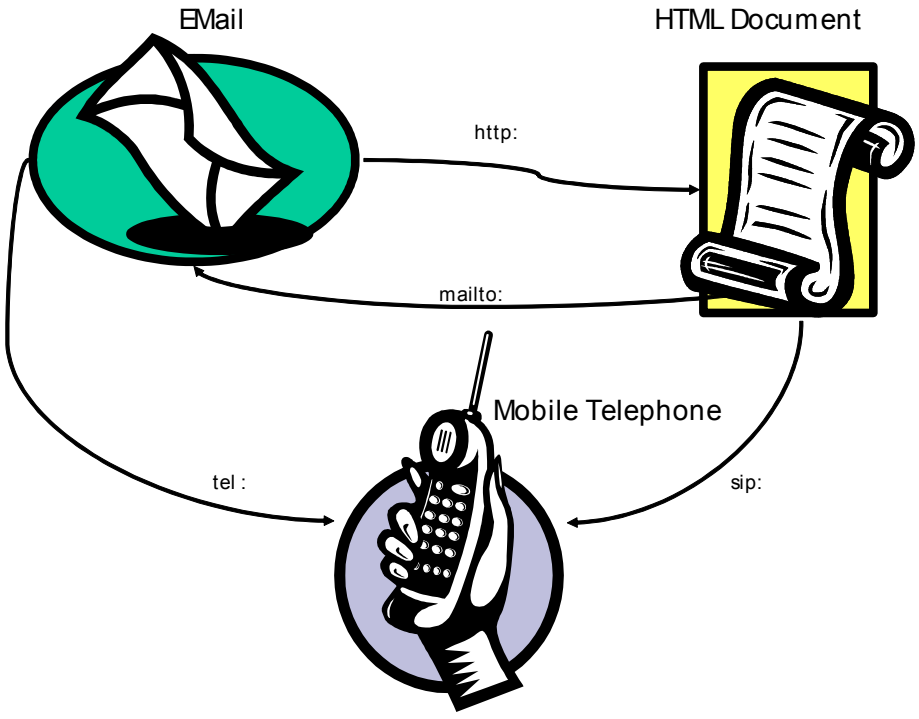

Figure 1: Scheme Interrelationships

4.1 URI Schemes

The specification and support for URI schemes has evolved over time and, while still not a completely "normative science", has converged in usage and specification. A number of web documents describe current best practices when specifying [RFC3986][RFC3305][RFC4395] registering [RFC4395] and applying [W3CWEBARCH]URI schemes. This section will define a set of common URI scheme technical requirements and define a minimum level of supported syntax and semantics for a set of OMA supported schemes.

4.2 General Scheme Support (Normative)

Absolute URIs are made up of a scheme part followed by a ":" (colon), and then a hierarchical or opaque part [RFC3986] (Appendix A.)

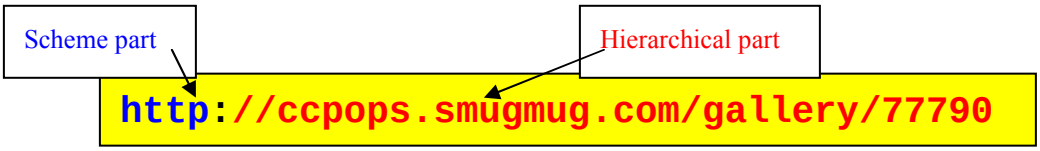

Figure 2: URI Part Names

URI schemes define a specific URI's hierarchical or opaque part semantics and processing within the framework of standard URI syntax [RFC3986]. In general there will be specialized client software required to handle each supported scheme. As a matter of convention and best practices, individual URI schemes tend to be processed by particular user agents. Thus it is convenient to think of each URI scheme as having a "logical user agent" to support client semantics and processing.

Client devices MUST support all referenced URI schemes that correspond to client supported "logical user agents".

User agents supporting end-user interaction SHOULD support URI syntax and processing.

Where the identification of a URI is explicit in the language syntax, such as the <a> anchor in XHTML, a user agent MUST support URI scheme processing.

Where the identification of a URI is not explicit in the syntax of the surrounding content, a user agent SHOULD identify and support all embedded strings that conform to specified URI syntax.

Strings such as embedded telephone numbers, addresses etc. that do not conform to URI syntax MAY be supported, if the system is able to identify such strings in situ, and assign a conforming URI to that string.

URIs, containing a supported URI scheme, MUST be processed in the same manner independent of the context in which the URI is found.

URIs which may be interpreted in multiple ways SHOULD present all appropriate alternatives to the user. (See 4.3 for discussion of ambiguous URIs.)

4.3 URI Scheme Semantics (Informative)

URIs and URI schemes are specified by a combination of Internet RFCs and W3C Architecture Documents. URIs represent a unique, addressable entity in the internet, and URI schemes represent the "scheme" in which the addressable entity makes sense. The goal of using a scheme based URI inside of some application context is to identify a target resource, perhaps to specify the context in which that resource will be used, and perhaps to specify an action for the target. This multiple role has led to ambiguities and confusion when attempting to use scheme based URIs as application directives when interworking amongst multiple applications.

In order to define an unambiguous application directive, three elements are needed:

- A target (**ID**) – an unambiguous identification string for an internet resource (e.g. E.164 telephone number, domainname, URL)

- An application (**L7**) - the application or user agent that can process data associated with a URI scheme.
- A protocol or network access approach (**NA**) – the protocol that is associated with a URI scheme.

This "triple" (heretofore known as the "scheme triple") defines a set of required information which unambiguously defines the intent and target of a URI. Optionally a URI may contain key/value pairs that further define application interaction. But these are specific to particular URI schemes and applications.

Complication and confusion have surrounded the use of URIs as applications directives due to the overloading of meaning on URI schemes and implied assumptions that are associated with the historical use of particular URI schemes. For example, up until recently the use of the "tel:" URI scheme implied the making of a telephone call. The scheme was acting as both the target ID, and the application definition. And since, for many years, there was only one way to make a phone call, the scheme also implied the network access approach to use. These days, there are many ways to make a telephone call (defined as making a connection to a telephony endpoint), and there are many applications that may be used. It has become necessary to "de-conflate" the various implied meanings of the "tel:" URI scheme. This is spelled out clearly in [RFC3966].

Because of usage and historical assumptions that are still in effect, the identification of the operating "scheme triple" is not always so straight forward. For example, the "http:" scheme clearly defines a target in the URL, and "http" part clearly defines the protocol (network access) to use. But what about the application part of the scheme triple? Some might think that the application implied is a web browser, but it is possible to point to URLs of all kinds. In reality, the application is implied by the MIME type of the referenced target. That is why all http stack implementations have a dispatch functionality that looks at the content type and sends the content to an appropriate application based on MIME type. This points to the final and perhaps most important aspect of using scheme based URIs. The binding of a URI to an action may occur at different points during the evaluation and execution of the URI, depending upon when all of the details of the "scheme triple" are known.. Early binding makes content developer intent clear. This can only happen when the "scheme triple" is complete at authoring time. Late binding, at the potential cost of ambiguous processing, provides flexibility in handling, allowing for multiple use case scenarios to be supported with a single syntax, and is more "future proof", not requiring the explicit listing of all relevant services.

Incomplete "scheme triples"simply represent the fundamental example of delayed binding. Using our "tel:" example again, if we receive a URI **tel:+35843256789** according RFC3966, this is a telephony endpoint (a target) only. So we cannot say what actions are being requested by this URI. It might be legitimate to put this target into our phone book if we can associate it with other information relevant to us. It might be possible to make a voice call to this endpoint, although we do not know the network access approach to be take (pstn, sip, H.323 etc.). Or perhaps this endpoint is associated with some application service such as faxing, video telephony or something as yet undefined. Binding the missing parts of the scheme triple may be accomplished in many ways. For example, IETF has defined the use of ENUM for identifying the applications available to a telephony endpoint. By using the telephony ID as an index to DNS, it is possible to return the services supported by the telephony endpoint. Sometimes endpoint services can be inferred by context. The inclusion of a "tel:" URI linked to the FAX part of a vCard, clearly implies that the endpoint is doing fax service.

The problem with much of the late binding of URIs to actions that takes place in the internet, is that it is either contextual, and as such difficult to automatically identify an appropriate service, or the late binding mechanism is not universally standardized (MIME types for http, and sip negotiation being notable exceptions).

The requirement for using a scheme based URI as an embedded content directive is now clear. All parts of the "scheme triple" must be bound at content authoring time, to ensure that the exact intent of the content author is met. Unbound elements at the time of request increase the range of acceptable client actions and may be desirable in certain circumstances.

In the following table we map out the "scheme triples" for all of the relevant URI schemes that are being considered. Schemes themselves may be considered to have a key orientation or class. For example "http:" clearly locks in the network access (protocol) to be used, so it is class NA (network access), whereas "tel:" explicitly defines a telephony target or ID, so it is of class ID. Due to the historical nature of some of the schemes, they may actually fit into multiple classes. Schemes that are in multiple classes tend to have tightly defined semantics, but less universal utility.

URI Scheme	Scheme Class	RFCs	ID	NA	L7
http:	NA	2616	URL	http	Client: Returned MIME type (late binding) Server: parameters
sip:	NA	3261	E.164 # or domainname	sip	Feature tags as defined in [RFC3840] Parameters (when available) User capabilities negotiation (late binding)
tel:	ID	3966	E.164 #	;call-type= (Vodaphone proposal)	ENUM if available (late binding)
mailto:	L7	2368	domainname	Implied historical linkage to smtp	email
sms:	NA,L7		E.164 #	sms bearer	SMS (short message service)
mmsto:	L7		E.164 # or domainname	http (specified)	MMS (Multi-media messaging)

URIs with incompletely defined “scheme triples” are very useful. Because they do not constrain all of the dimensions of the URI resource being identified, they are valid for multiple uses. As an example, if we are presented with the URI sip:bob@oma.org, we may have a number of services available to interact with “Bob”. A terminal device may present a menu of options, based on the SIP services that it can support. As mentioned above, tel:+358123456789 is simply a telephony endpoint. A terminal device could present a large number of applications and services that could use this telephone number as a target. Of course, we can also put the number in our telephone book.

5. URI Schemes

This section describes the normative application of the set of OMA supported URI schemes. The intent is that this small set of URI schemes is applied wherever applicable, across OMA user agents. The choice of the current set of specified schemes is based upon the following criteria:

- Applicability to logical user agents typically found in the mobile environment
- Ubiquity in the Internet at large
- Standards status – preference is given to those schemes that are registered with IESG or which are registered with the IETF through [RFC4395].

The user agent MUST handle character encoding according to [RFC3986], section 2.

TO guarantee correct interpretation of a URI, all URIs MUST be constructed as URL-encoded strings. To generate safe URIs one MUST NOT use unsafe octets.

```
; the actual value escaped to use only safe characters by replacing
; any unsafe-octet with its hex-escape
safe-char = ALPHA / DIGIT / "+" / "-" / "." / "%" / "_"
unsafe-octet = %x00-2A / %x2C / %x2F / %x3A-40 / %x5B-60 / %x7B-FF
hex-escape = "%" 2HEXDIG ; value of octet as hexadecimal value
```

5.1 http:

The “http:” scheme MUST be supported by clients whenever there is an http stack implemented and/or a logical browsing user agent present.

Table 1 - http:

Description	This URI scheme specifies an internet resource that is accessible via the Hypertext Transport Protocol.
Common Usage	http://www.myprofile.foo
Logical User Agent	Browser/content dispatch
Standardization Status	IETF Standards Track

5.1.1 Syntax

The syntax for the http: URI is as defined in [RFC2616] Section 3.2

```
http_URL = "http:" "//" host [ ":" port ] [ abs_path [ "?" query ]]
```

5.1.2 Semantics

The general semantics for the “http:” URI are defined as in [RFC2616]. Further conformance restrictions regarding resource retrieval in the mobile environment can be found in [WAESPEC] Section 7.2.1.

5.2 https:

The “https:”scheme MUST be supported by clients whenever there is a secure http stack implemented and/or logical browsing user agent present.

Table 2 - https:

Description	This URI scheme specifies an internet resource that is accessible via a secure end-to-end Hypertext Transport Protocol connection
Common Usage	https://www.myprofile.foo
Logical User Agent	Browser/content dispatch
Standardization Status	IETF Standards Track

5.2.1 Syntax

The syntax for the https: URI is as defined in [RFC2818] Section 2.4

```
https_URL = "https:" "//" host [ ":" port ] [ abs_path [ "?" query ]]
```

5.2.2 Semantics

The semantics of the URI are the same as http: with the added promise of end-to-end security for the payload that travels via the http protocol. That security is provided either by SSL or TLS. Further conformance restrictions regarding resource retrieval and security concerns in the mobile environment can be found in [WAESPEC] Section 7.2.2.

5.3 mailto:

The “mailto:” scheme MUST be supported by clients whenever there is an email logical user agent present.

Table 3 - mailto:

Description	The “mailto:” URL scheme is used to designate the Internet mailing address of an individual or service. In its simplest form, a “mailto:” URL contains an Internet mail address. For greater functionality, because interaction with some resources may require message headers or message bodies to be specified as well as the mail address, the mailto URL scheme is extended to allow setting mail header fields and the message body. (Section 1. [RFC2368])
Common Usage	mailto:ajm5634@myisp.com mailto:afriend@gmail.com?subject=We%20should%20talk.&body=Hi%20there.
Logical User Agent	email composer/mailer
Standardization Status	IETF Registered, standardized [RFC2368]

5.3.1 Syntax

Support for the"mailto:" URI scheme follows the syntax defined in [RFC2368]:

Given the syntax for mailto:

```
mailtoURL  =  "mailto:" [ to ] [ headers ]
to         =  #mailbox
headers    =  "?" header *( "&" header )
header     =  hname "=" hvalue
hname      =  *urlc
hvalue     =  *urlc
```

All hnames are case sensitive and MUST be lowercase (i.e. "to","subject","body","cc").

The following clarifications and exceptions to [RFC2368] apply.
The following hname/hvalue pairs MUST be supported:

Subject – The "**subject**" header MUST be supported.

Body– The "**body**" header MUST be supported.

The user-agent SHOULD be architected to avoid any intrinsic limits to the number of characters supported in the body of the "mailto:" URI scheme.

Note: Effective limits may be present because some user-agents still have difficulties in supporting more than 256 characters.

Cc – The "**cc**" header MUST be supported.

To - The "**to**" header MUST be supported (see [RFC2368] for relationship between the"to" header and the to field. The example mailto:?to=annie@hotmail.com is legal but uncommon.)

Note: Multiple address fields such as in a "to" or "cc" are specified by [RFC2822] as comma separated lists. These commas MUST be URL-encoded (","=%2C) to guarantee a safely parsable URI string.

Logical user agents MUST display the header and value to the end-user. That value MUST be able to be overridden or edited by the end-user.

Due to security concerns, the following hname/hvalue pair(s) MUST NOT be supported:

From – The "**from**" header, if encountered, MUST be ignored, and the true sender used.

Other headers (hnames) MAY be supported under the following conditions:

Logical user agents MUST display the header and value to the end-user.

Supported hname/value pairs MUST be able to be overridden or edited by the logical user agent and the end-user.

Headers that are deemed, by the logical user agent, to represent any security concern SHOULD be ignored. Examples may include headers that allow for the spoofing of identities or return paths.

Any hname/value pairs (with the exception of "to","subject","body","cc"). MAY be ignored if they are not recognized.

5.3.2 Semantics

Support for the "mailto:" URI scheme follows the semantics defined in [RFC2368] with the following clarifications or exceptions:

The user agent handling the"mailto:" URI MUST invoke an end-user interactive action. It MUST NOT cause the issuance of email without consent and viewing by the end-user. (Note: Section 7 [RFC2368] is a SHOULD).

The logical email composition user agent MUST ignore the "**`from`**" field if specified.

Whitespace treatment inside "`body`" and "`subject`" MUST be passed onto the logical user agent unchanged. URI encoded format (e.g. spaces are encoded as %20) MUST not be interpreted.

5.4 tel:

The "tel:" scheme MUST be supported by clients whenever there is a logical user agent present that identifies and/or communicates with other communications endpoints using an E.164 telephone number.

Table 4 - tel:

Description	The "tel:" scheme describes resources identified by telephone numbers. It implies the service (or action) of interacting with a telephony endpoint identified by a telephone number, although not how that telephone number is reached or the exact nature of the service.
Common Usage	tel:+17815551234
Logical User Agents	telephony services, fax services, sip services, data services
Standardization Status	[RFC3966] Standards Track

5.4.1 Syntax

Support for the "tel:" URI scheme follows the syntax defined in [RFC3966],

Given the syntax for tel: is defined as:

```
telephone-uri        = "tel:" telephone-subscriber
telephone-subscriber = global-number / local-number
global-number        = global-number-digits *par
local-number         = local-number-digits *par context *par
par                  = parameter / extension / isdn-subaddress
isdn-subaddress      = ";isub=" 1*uric
extension            = ";ext=" 1*phonedigit
context              = ";phone-context=" descriptor
descriptor           = domainname / global-number-digits
global-number-digits = "+" 1*phonedigit
local-number-digits  = 1*phonedigit-hex
domainname           = *( domainlabel "." ) toplabel [ "." ]
domainlabel          = alphanum / alphanum *( alphanum /"-" ) alphanum
toplabel             = ALPHA / ALPHA *( alphanum / "-" ) alphanum
parameter            = ";" pname ["=" pvalue ]
pname                = 1*( alphanum / "-" )
pvalue               = 1*paramchar
paramchar            = param-unreserved / unreserved / pct-encoded
unreserved           = alphanum / mark
mark                 = "-" / "_" / "." / "!" / "~" / "*" /
                       "'" / "(" / ")"
pct-encoded          = "%" HEXDIG HEXDIG
param-unreserved     = "[" / "]" / "/" / ":" / "&" / "+" / "$"
phonedigit           = DIGIT / [ visual-separator ]
phonedigit-hex       = HEXDIG / "*" / "#" / [ visual-separator ]
visual-separator     = "-" / "." / "(" / ")"
alphanum             = ALPHA / DIGIT
reserved             = ";" / "/" / "?" / ":" / "@" / "&" /
                       "=" / "+" / "$" / ","
```

```
uric                = reserved / unreserved / pct-encoded
```

The following optional parameter is defined to act as a hint as to the intended usage for the tel: URI instance:

```
pname                = "call-type"
pvalue         ="voice"/"fax"/"modem"/"pstn"/"video"/"sms"/"mms"
```

Example : tel:+4446543345;call-type=voice

The above example URI only suggests that the application performs a voice-call. However, this URI does not indicate a contractual obligation on the part of the application, only a hint as to the expectations on the part of the content creator. The end user is at liberty to accept, reject or select an alternative.

As noted in ([RFC3966] Section 5.4) this parameter may be ignored, as it is optional. The terminal device SHOULD use best effort to execute the requested operation to the requested telephony endpoint.

5.4.2 Semantics

Support for the "tel:" URI scheme follows the semantics defined in [RFC3966] .

The URI simply describes a unique telphony endpoint, and does not define the type of service that may be associated with the termination point defined by the URI. It may include voice, video, data or fax. The usage of a particular local logical user agent and the establishment of a particular type of connection is the responsibility of the device.

The URI is not the "dial-string" that may be required to effect a connection to a particular terminal service. The responsibility of translating the "tel:" URI to a "dial-string" is owned by the logical user agent. For example, if there are required pauses, or digits required to navigate out through a PBX, application of that transformation to the URI is the responsibility of the logical user agent. Local telephony context may be specified as defined in [RFC3966] Section 4.

Note: This is the major change between the obsolete RFC2806 and [RFC3966], is that the assignment of a particular action has been removed from the specification. Hence the problem of having to define an ever growing list of agents and protocols which rely on telephone number semantics has been avoided.. That is why the "fax:" and "modem:" URI schemes have been removed.

The assignment of an action to be associated with a "tel:" URI instance is out of scope for this document, but it is assumed that all available means will be used to provide end-users with a logical, ordered set of actionable alternatives. These include, but are not limited to, URI parameter usage hint, document context, past user history, DNS enumeration and other mechanisms.

5.5 sip:

The "sip:" URI scheme MUST be supported by clients whenever there is one or more logical user agents present that support SIP based applications.

Table 5 - sip:

Description	The "sip:" URI scheme identifies a communications resource that is addressable through the Session Initiation Protocol (SIP). It is used by many different applications, and in its simplest form assumes that the application parameters for interaction will be negotiated as part of the SIP interaction.
Common Usage	sip:billyboy@yourdomain.com
Logical User Agent	Since SIP is a protocol, many different logical user agents may act as the SIP logical user agent.
Standardization Status	Standards Track - [RFC3261]

5.5.1 Syntax

Full BNF syntax for sip: and sips: is found in [RFC3261] Section 25.1. The following incomplete BNF illustrates where parameters are used in sip:.

```
SIP-URI          =  "sip:" [ userinfo ] hostport
                    uri-parameters [ headers ]
userinfo         =  ( user / telephone-subscriber ) [ ":" password ] "@"
user             =  1*( unreserved / escaped / user-unreserved )
user-unreserved  =  "&" / "=" / "+" / "$" / "," / ";" / "?" / "/"
password         =  *( unreserved / escaped /
                    "&" / "=" / "+" / "$" / "," )
hostport         =  host [ ":" port ]
host             =  hostname / IPv4address / IPv6reference
hostname         =  *( domainlabel "." ) toplabel [ "." ]
domainlabel      =  alphanum
                    / alphanum *( alphanum / "-" ) alphanum
toplabel         =  ALPHA / ALPHA *( alphanum / "-" ) alphanum

...
other-param       =  pname [ "=" pvalue ]
pname             =  1*paramchar
pvalue            =  1*paramchar
paramchar         =  param-unreserved / unreserved / escaped
param-unreserved  =  "[" / "]" / "/" / ":" / "&" / "+" / "$"

headers         =  "?" header *( "&" header )
header          =  hname "=" hvalue
hname           =  1*( hnv-unreserved / unreserved / escaped )
hvalue          =  *( hnv-unreserved / unreserved / escaped )
hnv-unreserved  =  "[" / "]" / "/" / "?" / ":" / "+" / "$"

...
```

[RFC3840] extends the Contact header field. In particular, it allows for the Contact header field parameters to include feature-param. Feature-param is a feature parameter (also known as a feature tag) that describes a feature of the UA associated with the URI in the Contact header field. Feature parameters are identifiable because they either belong to the well known set of base feature tags, or they begin with a plus sign.

```
feature-param    =  enc-feature-tag [EQUAL LDQUOT (tag-value-list
                    / string-value ) RDQUOT]
```

```
enc-feature-tag  =  base-tags / other-tags
base-tags        =  "audio" / "automata" /
                    "class" / "duplex" / "data" /
                    "control" / "mobility" / "description" /
                    "events" / "priority" / "methods" /
                    "schemes" / "application" / "video" /
                    "language" / "type" / "isfocus" /
                    "actor" / "text" / "extensions"
other-tags       =  "+" ftag-name
ftag-name        =  ALPHA *( ALPHA / DIGIT / "!" / "'" /
                    "." / "-" / "%" )
```

5.5.2 Semantics

When the feature tag “+g.poc.talkburst” or” +g.poc.groupad” is present as part of the sip URI, the URI MUST be interpreted as a request to a user agent capable of handling POC requests ([[OMATSPOCCP]] Section 6).

When the feature tag “+g.oma.sip-im” is present as part of the sip URI, the URI MUST be interpreted as a request to a user agent capable of handling OMA Instant Messaging requests ([[OMATSSIMPLE]] Section 6).

5.6 sips:

The “sips:” URI scheme MUST be supported by clients whenever there is a logical user agent present which supports a secure SIP based application.

Table 6 - sips:

Description	The “sips:” URI scheme identifies a secure communications resource that is addressable through the Session Initiation Protocol (SIP). It is used by many different applications, and in its simplest form assumes that the application parameters for interaction will be negotiated as part of the SIP interaction.
Common Usage	sips:jeanysue@mydomain.com
Logical User Agent	Since SIP is a protocol, many different logical user agents may act as the SIP logical user agent.
Standardization Status	Standards Track - [RFC3261]

5.6.1 Syntax

```
SIPS-URI         =  "sips:" [ userinfo ] hostport
                    uri-parameters [ headers ]
```

…

See section 5.5.1

5.6.2 Semantics

The “sips:” URI scheme implies an end-to-end secure connection as defined in [RFC3261].

See section 5.5.2.

5.7 sms:

The “sms:” URI scheme MUST be supported by clients whenever there is a logical user agent present that supports SMS messaging.

Table 7 - sms:

Description	The “sms:” URI scheme is used to designate the global telephony address of an individual or service, and implies the delivery of content using SMS messaging protocols and formats.
Common Usage	 Send an SMS to England!
Logical User Agent	sms composer/sender
Standardization Status	IETF Draft [SMSSCHEME]

5.7.1 Syntax

Syntax of the “sms:” URI scheme is described by [SMSSCHEME].

```
sms-uri               =  scheme ":" scheme-specific-part
scheme                =  "sms"
scheme-specific-part  =  1*( sms-recipient ) [ sms-body ]
sms-recipient         =  gstn-phone sms-qualifier
                         [ "," sms-recipient ]
sms-qualifier         =  *( smsc-qualifier / pid-qualifier )
smsc-qualifier        =  ";smsc=" SMSC-sub-addr
pid-qualifier         =  ";pid=" PID-sub-addr
sms-body              =  "?body=" *urlc
```

5.7.2 Semantics

Semantics of the “sms:” URI scheme are described by [SMSSCHEME].

SMS requests made through URI syntax MUST NOT cause the automatic generation and transmission of an SMS message. End user acceptance of an SMS send action MUST be required.

5.8 mmsto:

Note: this scheme will be registered to OMA using the IETF scheme registration facility as defined in [RFC4395]

The “mmsto:” URI scheme MUST be supported by clients whenever there is a logical user agent present that supports MMS messaging.

Table 8 - mmsto:

Description	The “mmsto:” URL scheme is used to designate the global telephony address or Internet mailing address of an individual or service, and implies the delivery of content using MMS messaging protocols and formats.
Common Usage	mmsto:+17913331234?cc=fred@picdom.com&subject=yo%20there

Logical User Agent	mms composer/sender
Standardization Status	Owned by OMA

5.8.1 Syntax

The "mmsto:" URI scheme is intended for addressing an MMS message to certain recipients. The functionality is quite similar to that of the "mailto:" URL scheme [RFC2368], that can be used with a comma-separated list of email addresses. In both cases, additional message parameters (such as the subject and message body) may be provided.

How the MMS message is composed and subsequently sent to the MMS Proxy-Relay is outside the scope of this specification. MMS messages can be sent over the GSM air interface, by using a modem and a suitable protocol, or by accessing services over other protocols, such as a Web service for sending MMS messages.

The "mmsto:" URI is case-insensitive. The syntax of an "mmsto:" URI is formally described as follows, where the base syntax is taken from [RFC3986]:

```
mmsto-uri                = mmsto-scheme ":" scheme-specific-part
mmsto-scheme             = “mmsto:”
scheme-specific-part     = [ to ] [ headers ]
to                       =  1*( mms-address )
headers                  = "?" header *( "&" header )
header                   = hname "=" hvalue
hname                    = *urlc
hvalue                   = *urlc
mms-address              = rfc2822-addr
                           / dev-addr
                           /asc-addr
                           / ["," mms-address]
dev-addr                 = gstn-phone ["/TYPE=PLMN"]
                           / ipv4 ["/TYPE=IPv4"]
                           / ipv6 ["/TYPE=IPv6"]
rfc822-addr              = mailbox ["/TYPE=rfc2822"]
asc-addr                 = *urlc
```

The "mms-address" is defined by the MMS Addressing model as specified in ([OMAMMSENC] Section 8). A valid MMS recipient address can be either an E.164 (MSISDN) , an RFC2822 address or a short code ascii address.

The "hname" and "hvalue" are encodings of an MMS informational element name and value, unless a mapping already exists to [RFC2822], as specified in [OMAMMSENC].

The "gstn-phone" is as specified in [RFC3601] allowing global as well as local telephone numbers. Note: local telephone numbers can be problematic and should not be used, unless a local to global mapping is available.

The "mailbox" is as specified in [RFC2822]. This means that it consists of zero or more comma-separated mail addresses, possibly including "phrase" and "comment" components.

The "asc-addr" is an alphanumeric short code that is interpreted by the MMS infrastructure as specified in [OMAMMSENC].

All URI reserved characters in "to" must be url-encoded: in particular parentheses (%28,%29), slashes (%2F), commas (%2C), and the percent sign "%" (%25), which commonly occur in "mailbox" syntax.

MMS messages may contain a variety of headers as specified in [OMAMMSENC]. The "hvalue" and "hvalue" are encodings of MMS information element name and values, where the information elementname has been hyphenated.

The following header(s) MUST be supported:

```
Body         = "body" "=" unstructured
To-header    = "to" "=" 1*( mms-address )
Cc-header    = "cc" "=" 1*( mms-address )
```

```
Subject-header = "subject" "=" unstructured
```

The following headers MAY be supported:

```
Message-class = "Message-Class" "=" ( Class-identifier | quoted-string
)
Class-identifier = "Personal" | "Advertisement" | "Informational" |
"Auto"
Expiry-value = "Expiry" "=" ( HTTP-date | delta-seconds )
Delivery-report = "Delivery-Report" "=" ( "Yes" | "No" )
Priority = "Priority" "=" ( "Low" | "Normal" | "High" )
Sender-visibility = "Sender-Visibility" "=" ( "Hide" | "Show" )
Read-reply = "Read-Reply" "=" ( "Yes" | "No" )
```

All URI reserved characters MUST be encoded. 8-bit characters in “mmsto:” URLs are forbidden. MIME encoded words (as defined in [RFC2047]) are permitted in header values, but not for any part of a "body" hname.

Within "mmsto:" URIs, the characters "?", "=", "&" are reserved.Because the "&" (ampersand) character is reserved in HTML, any "mmsto:" URI that contains an ampersand must be spelled differently in HTML than in other contexts. An "mmsto:" URI that appears in an HTML document must use & instead of "&".

Also note that it is legal to specify both "to" and a "hname" whose value is "to". That is,

mmsto:addr1%2C%20addr2 is equivalent to

mmsto:?to=addr1%2C%20addr2 is equivalent to

mmsto:addr1?to=addr2

5.8.2 Semantics

An "mmsto:" URI identifies an "internet resource" corresponding to the MMS mailbox specified in the address. When additional headers are supplied, the resource designated is the same address, but with an additional profile for accessing the resource.

In current practice, resolving URIs and URLs such as those in the "http" scheme causes an immediate interaction between user agents and a host running an interactive server. In contrast, the "mmsto:" URI has similar semantics as the "mailto" URL -- resolving the "mmsto:" URI does not cause an immediate interaction. Rather, the user agent creates a message to the designated address with the various header fields set as default. The user can edit the message, send this message unedited, or choose not to send the message.

The following list describes the steps for processing an "mmsto:" URI by an MMS user agent:

1. The user agent MUST extract the plmn-addr, rfc822-addr, or short code of each "mms-address". The user agent SHOULD ignore recipients with invalid syntax.

2. The user agent MUST extract each header ("hname" and "hvalue"pair) in order.

3. The user agent SHOULD NOT create the message if any of the headers are considered to be security risks. The user agent MAY also choose to create a message with only a subset of the headers given in the URI. Only the mandatory headers (to", "cc","subject" and "body") are considered safe. The creator of an "mmsto:" URI cannot expect the resolver of a URI to understand more than the "to", "cc","subject" and "body" headers. Clients that resolve "mmsto:" URIs into multimedia messages MUST be able to correctly create MMS-compliant messages using the "subject" and "body" headers.

4. The user agent MUST provide some means for an end-user to verify and modify the URI parameters passed into a user agent as part of the "mmsto:" URI, either by implementing this itself, or accessing a service or application providing message composition.

5. The user agent MUST require end-user acknowledgement to send an MMS.

5.9 Unsupported URI Schemes

Schemes not supported by a particular client device MUST be treated as any other unknown URL, and MUST NOT cause the device to fail.

Dynamic registration of URI schemes is a logical way to support the extension of support to new schemes but is not mandated by this specification.

Processing of unregistered schemes is a device concern, and out of scope for this specification. However, the device SHOULD notify the end-user when it is unable to process a URI request.

5.10 Legacy URI Schemes (Informative)

A number of URI schemes have been defined by OMA or their antecedent organizations. Unless explicitly called out in this chapter, this specification makes no statement about their continued use or about their use outside of the explicit specification context in which they are described.

These schemes are listed in [WAESPEC] Section 7.2.3.

5.11 Application Registration (Informative)

A facility to manage the bindings between a URI scheme and its "logical user agent(s)" is a convenient way to manage "logical user agents" support for particular URI schemes. However, no particular approach is mandated by this specification.

5.12 Interpretation of a URI (Informative)

As shown in section 4.3, URIs, at the point they are interpreted by a system, may be either, unambiguous, leading to a clear action or ambiguous, representing useful information that may be put to any number of uses. It is expected that support for ambiguous URIs will result in the presentation of a set of legitimate alternative uses for a URI. As an example a tel: URI with an E.164 phone number, and no other qualifications could be used in:

- Making a PSTN telephone call
- As an entry in a phonebook
- As an SMS address
- As an MMS address
- As a POC endpoint
- As a SIP VOIP address
- As a Video Telephony endpoint

Typically alternatives would be presented as a menu, but no particular presentation is mandated.

Table 9. Typical applications using URIs

	tel:	sms:	mmsto:	sip: / sips:	http:/ https:	mailto:
Telephony	✓			✓		
Phonebook	✓	✓	✓	✓		✓

SMS	✓	✓				
MMS	✓		✓			
Browser					✓	
Email						✓
PoC	✓			✓		
IM				✓		
VoIP	✓			✓		
Video Telephony	✓			✓		
Multimedia			✓		✓	

It is the responsibility of the content author to provide an unambiguous URI, if a particular action is expected.

Appendix A. Change History (Informative)

A.1 Approved Version 1.0 History

Reference	Date	Description
OMA-TS-URI_Schemes-V1_0	26 Jun 2008	Status changed to Approved by TP: OMA-TP-2008-0256- INP_URI_Schemes_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 SCR for URIScheme Client

Item	Function	Reference	Status	Requirement
URIS-C-001	Support for all URI schemes that correspond to client supported "logical user agents"	Section 4.2	M	(URIS-C-002 OR URIS-C-005 OR URIS-C-008 OR URIS-C-011 OR URIS-C-014 OR URIS-C-017 OR URIS-C-020 OR URIS-C-023) AND URIS-C-026
URIS -C-002	Support for the http: URI scheme	Section 5.1	O	URIS-C-003 AND URIS-C-004
URIS -C-003	Support for http: syntax	Section 5.1.1	O	
URIS -C-004	Support for http: semantics	Section 5.1.2	O	
URIS -C-005	Support for the https: URI scheme	Section 5.2	O	URIS-C-006 AND URIS-C-007
URIS -C-006	Support for https: syntax	Section 5.2.1	O	
URIS -C-007	Support for https: semantics	Section 5.2.2	O	
URIS -C-008	Support for the mailto: URI scheme	Section 5.3	O	URIS-C-009 AND URIS-C-010
URIS -C-009	Support for mailto: syntax	Section 5.3.1	O	
URIS -C-010	Support for mailto: semantics	Section 5.3.2	O	
URIS -C-011	Support for the tel: URI scheme	Section 5.4	O	URIS-C-012 AND URIS-C-013
URIS -C-012	Support for tel: syntax	Section 5.4.1	O	
URIS -C-013	Support for tel: semantics	Section 5.4.2	O	
URIS -C-014	Support for the sip: URI scheme	Section 5.5	O	URIS-C-015 AND URIS-C-016
URIS -C-015	Support for sip: syntax	Section 5.5.1	O	
URIS -C-016	Support for sip: semantics	Section 5.5.2	O	
URIS -C-017	Support for the sips: URI scheme	Section 5.6	O	URIS-C-018 AND URIS-C-019
URIS -C-018	Support for sips: syntax	Section 5.6.1	O	
URIS -C-019	Support for sips: semantics	Section 5.6.2	O	
URIS-C-020	Support for the mmsto: URI scheme	Section 5.8	O	URIS-C-021 AND URIS-C-022
URIS-C-021	Support for mmsto: syntax	Section 5.8.1	O	
URIS-C-022	Support for mmsto: semantics	Section 5.8.2	O	
URIS-C-023	Support for the sms: URI scheme	Section 5.7	O	URIS-C-024 AND URIS-C-025
URIS-C-024	Support for sms: syntax	Section 5.7.1	O	
URIS-C-025	Support for sms: semantics	Section 5.7.2	O	
URIS-C-026	Handling of unsupported schemes	Section 5.9	M	