URI Scheme for Global System for Mobile Communications (GSM)
Short Message Service (SMS)

Abstract

   This memo specifies the Uniform Resource Identifier (URI) scheme
   "sms" for specifying one or more recipients for an SMS message. SMS
   messages are two-way paging messages that can be sent from and
   received by a mobile phone or a suitably equipped networked device.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc5724.

POSTSCRIPT 1019

Table of Contents

1. Introduction

1.1. What is GSM?

GSM (Global System for Mobile Communications) is a digital mobile
phone standard that is used extensively in many parts of the world.
First named after its frequency band around 900 MHz, GSM-900 has
provided the basis for several other networks utilizing GSM
technology, in particular, GSM networks operating in the frequency
bands around 1800 MHz and 1900 MHz. When referring to "GSM" in this
document, we mean any of these GSM-based networks that operate a
short message service.

1.2. What is SMS?

The Short Message Service (SMS) [SMS] is an integral part of the GSM
network technology. It has been very successful and currently is a
major source of revenue for many GSM operators. SMS as a service is
so successful that other Global Switched Telephone Network (GSTN)
technologies have adopted it as well, in particular, the Integrated
Services Digital Network (ISDN). Because of this development, this
memo uses the term "SMS client" to refer to user agents that are able
to send and/or receive SMS messages.

1.2.1. SMS Content

GSM SMS messages are alphanumeric paging messages that can be sent to
and from SMS clients. SMS messages have a maximum length of 160
characters (7-bit characters from the GSM character set [SMS-CHAR]),
or 140 octets. Other character sets (such as UCS-2 16-bit
characters, resulting in 70-character messages) MAY also be supported
[SMS-CHAR], but are defined as being optional by the SMS
specification. Consequently, applications handling SMS messages as
part of a chain of character-processing applications MUST make sure
that character sets are correctly mapped to and from the character
set used for SMS messages.

While the 160-character content type for SMS messages is by far the
one most widely used, there are numerous other content types for SMS
messages, such as small bitmaps ("operator logos") and simple formats
for musical notes ("ring tones"). However, these formats are
proprietary and are not considered in this memo.

SMS messages are limited in length (140 octets), and the first
versions of the SMS specification did not specify any standardized
methods for concatenating SMS messages. As a consequence, several
proprietary methods were invented, but the current SMS specification
does specify message concatenation. In order to deal with this

situation, SMS clients composing messages SHOULD use the standard
concatenation method based on the header in the TP-User Data field as
specified in the SMS specification [SMS]. When sending a message to
an SMS recipient whose support for concatenated messages is unknown,
the SMS client MAY opt to use the backwards-compatible (text-based)
concatenation method defined in the SMS specification [SMS].
Proprietary concatenation methods SHOULD NOT be used except in closed
systems, where the capabilities of the recipient(s) are always known.

1.2.2. SMS Infrastructure

SMS messages can be transmitted over an SMS client's network
interface using the signaling channels of the underlying GSTN
infrastructure, so there is no delay for call setup. Alternatively,
SMS messages may be submitted through other front-ends (for example,
Web-based services), which makes it possible for SMS clients to run
on computers that are not directly connected to a GSTN network
supporting SMS.

SMS messages sent with the GSTN SMS service MUST be sent as class 1
SMS messages, if the client is able to specify the message class.

1.2.2.1. SMS Centers

For delivery within GSTN networks, SMS messages are stored by an
entity called SMS Center (SMSC) and sent to the recipient when the
subscriber connects to the network. The number of a cooperative SMSC
must be known to the SMS sender (i.e., the entity submitting the SMS
message to a GSTN infrastructure) when sending the message (usually
the SMSC's number is configured in the SMS client and specific for
the network operator to which the sender has subscribed). In most
situations, the SMSC number is part of the sending SMS client's
configuration. However, in some special cases (such as when the SMS
recipient only accepts messages from a certain SMSC), it may be
necessary to send the SMS message over a specific SMSC. The scheme
specified in this memo does not support the specification of SMSC
numbers, so in case of scenarios where messages have to be sent
through a certain SMSC, there must be some other context establishing
this requirement or message delivery may fail.

1.2.3. Uniform Resource Identifiers

One of the core specifications for identifying resources on the
Internet is [RFC3986], specifying the syntax and semantics of a
Uniform Resource Identifier (URI). The most important notion of URIs
are "schemes", which define a framework within which resources can be
uniquely identified and addressed. URIs enable users to access
resources and are used for very diverse schemes, such as access

protocols (HTTP, FTP), broadcast media (TV channels [RFC2838]),
messaging (email [RFC2368]), and even telephone numbers (voice
[RFC3966]).

URIs often are mentioned together with Uniform Resource Names (URNs)
and/or Uniform Resource Locators (URLs), and it often is unclear how
to separate these concepts. For the purpose of this memo, only the
term URI will be used, referring to the most fundamental concept.
The World Wide Web Consortium (W3C) has issued a note
[uri-clarification] discussing the topic of URIs, URNs, and URLs in
detail.

1.2.4. SMS Messages and the Internet

One of the important reasons for the universal access of the Web is
the ability to access all information through a unique interface.
This kind of integration makes it easy to provide information as well
as to consume it. One aspect of this integration is the support of
user agents (in the case of the Web, commonly referred to as
browsers) for multiple content formats (such as HTML, GIF, JPEG) and
access schemes (such as HTTP, HTTPS, FTP).

The "mailto" scheme has proven to be very useful and popular because
most user agents support it by providing an email composition
facility when the user selects (e.g., clicks on) the URI. Similarly,
the "sms" scheme can be supported by user agents by providing an SMS
message composition facility when the user selects the URI. In cases
where the user agent does not provide a built-in SMS message
composition facility, the scheme could still be supported by opening
a Web page that provides such a service. The specific Web page to be
used could be configured by the user, so that each user could use the
SMS message composition service of his choice.

The goal of this memo is to specify the "sms" URI scheme so that user
agents (such as Web browsers and email clients) can start to support
it. The "sms" URI scheme identifies SMS message endpoints as
resources. When "sms" URIs are dereferenced, implementations MAY
create a message and present it to be edited before being sent, or
they MAY invoke additional services to provide the functionality
necessary for composing a message and sending it to the SMS message
endpoint. In either case, simply activating a link with an "sms" URI
SHOULD NOT cause a message to be sent without prior user
confirmation.

1.2.4.1. SMS Messages and the Web

SMS messages can provide an alternative to "mailto" URIs [RFC2368], or "tel" or "fax" URIs [RFC3966]. When an "sms" URI is activated, the user agent MAY start a program for sending an SMS message, just as "mailto" may open a mail client. Unfortunately, most browsers do not support the external handling of internally unsupported URI schemes in the same generalized way as most of them support external handling of content for media types that they do not support internally. Ideally, user agents should implement generic URI parsers and provide a way to associate unsupported schemes with external applications (or Web-based services).

The recipient of an SMS message need not be a mobile phone. It can be a server that can process SMS messages, either by gatewaying them to another messaging system (such as regular electronic mail), or by parsing them for supplementary services.

SMS messages can be used to transport almost any kind of data (even though there is a very tight size limit), but the only standardized data formats are character-based messages in different character encodings. SMS messages have a maximum length of 160 characters (when using 7-bit characters from the SMS character set), or 140 octets. However, SMS messages can be concatenated to form longer messages. It is up to the user agent to decide whether to limit the length of the message, and how to indicate this limit in its user interface if necessary. There is one exception to this; see Section 2.6.

1.2.4.2. SMS Messages and Forms

The Hypertext Markup Language (HTML) [HTML401] provides a way to collect information from a user and pass it to a server for processing. This functionality is known as "HTML forms". A filled-in form is usually sent to the destination using the Hypertext Transfer Protocol (HTTP) or email. However, SMS messages can also be used as the transport mechanism for these forms. Depending on the network configuration, the sender's telephone number may be included in the SMS message, thus providing a weak form of authentication.

1.3. Requirements Language

The capitalized 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].

2. The "sms" URI Scheme

   Syntax definitions are given using the Augmented BNF (ABNF) for
   syntax specifications [RFC5234].

2.1. Applicability

   This URI scheme provides information that can be used for sending SMS
   message(s) to specified recipient(s). The functionality is
   comparable to that of the "mailto" URI, which (as per [RFC2368]) can
   also be used with a comma-separated list of email addresses.

   The notation for phone numbers is taken from [RFC3966] and its
   Erratum 203 [Err203]. Appendix A provides a corrected syntax of the
   telephone number. Refer to that document for information on why this
   particular format was chosen.

   How SMS messages are sent to the SMSC or other intermediaries is
   outside the scope of this specification. SMS 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-based
   service for sending SMS messages. Also, SMS message service options
   like deferred delivery and delivery notification requests are not
   within the scope of this document. Such services MAY be requested
   from the network by the user agent if necessary.

   SMS messages sent as a result of this URI MUST be sent as class 1 SMS
   messages, if the user agent is able to specify the message class.

2.2. Formal Definition

   The URI scheme's keywords specified in the following syntax
   description are case-insensitive. The syntax of an "sms" URI is
   formally described as follows, where the URI base syntax is taken
   from [RFC3986]:

```
sms-uri        = scheme ":" sms-hier-part [ "?" sms-fields ]
scheme         = "sms"
sms-hier-part  = sms-recipient *( "," sms-recipient )
sms-recipient  = telephone-subscriber ; defined in RFC 3966
sms-fields     = sms-field *( "&" sms-field )
sms-field      = sms-field-name "=" escaped-value
sms-field-name = "body" / sms-field-ext ; "body" MUST only appear once
sms-field-ext  = 1*( unreserved )
escaped-value  = *( unreserved / pct-encoded ) ; defined in RFC 3986
```

Some illustrative examples using this syntax are given in
Section 2.5.

The syntax definition for <telephone-subscriber> is taken from
Section 5.1 of [RFC3966]. Please consider Erratum 203 [Err203] in
that specification. For the reader's convenience, Appendix A
contains a fixed syntax of the telephone number URI scheme, including
Erratum 203, but RFC 3966 (plus all applicable errata) is the
normative reference. The description of phone numbers in RFC 3966
(Section 5.1) states: "The 'telephone-subscriber' part of the URI
indicates the number. The phone number can be represented in either
global (E.164) or local notation. All phone numbers MUST use the
global form unless they cannot be represented as such. Numbers from
private numbering plans, emergency ("911", "112"), and some
directory-assistance numbers (e.g., "411") and other "service codes"
(numbers of the form N11 in the United States) cannot be represented
in global (E.164) form and need to be represented as a local number
with a context. Local numbers MUST be tagged with a 'phone-
context'."

This specification defines a single <sms-field>: "body". Extensions
to this specification MAY define additional fields. Extensions MUST
NOT change the semantics of the specifications they are extending.
Unknown fields encountered in "sms" URIs MUST be ignored by
implementations.

The "body" <sms-field> is used to define the body of the SMS message
to be composed. It MUST not appear more than once in an "sms" URI.
It consists of percent-encoded UTF-8 characters. Implementations
MUST make sure that the "body" <sms-field> characters are converted
to a suitable character encoding before sending, the most popular
being the 7-bit SMS character encoding, another variant (though not
as universally supported as 7-bit SMS) is the UCS-2 character
encoding (both specified in [SMS-CHAR]). Implementations MAY choose
to discard (or convert) characters in the <sms-body> that are not
supported by the SMS character set they are using to send the SMS
message. If they do discard or convert characters, they MUST notify
the user.

The syntax definition for <escaped-value> refers to the text of an
SMS where all <reserved> (as per [RFC3986]) characters in the SMS
text are percent-encoded, please refer to [RFC3986] for the
definitions of <unreserved> and <pct-encoded> and for details about
percent-encoding.

User agents SHOULD support multiple recipients and SHOULD make it
clear to users what the entire list of recipients is before
committing the user to sending all the messages.

2.3. Processing an "sms" URI

The following list describes the steps for processing an "sms" URI:

1. The phone number of the first <sms-recipient> is extracted. It
    is the phone number of the final recipient and it MUST be written
    in international form with country code, unless the number only
    works from inside a certain geographical area or a network. Note
    that some numbers may work from several networks but not from the
    whole world -- these SHOULD be written in international form.
    According to [RFC3966], all international numbers MUST begin with
    a "+" character. Hyphens, dots, and parentheses (referred to as
    "visual separators" in RFC 3966) are used only to improve
    readability and MUST NOT convey any other meaning.

2. The "body" <sms-field> is extracted, if present.

3. The user agent SHOULD provide some means for message composition,
    either by implementing this itself or by accessing a service that
    provides it. Message composition SHOULD start with the body
    extracted from the "body" <sms-field>, if present.

4. After message composition, a user agent SHOULD try to send the
    message first using the default delivery method employed by that
    user agent. If that fails, the user agent MAY try another
    delivery method.

5. If the URI contains a comma-separated list of recipients (i.e.,
    it contains multiple <sms-recipient> parts), all of them are
    processed in this manner. Exactly the same message SHOULD be
    sent to all of the listed recipients, which means that the
    message resulting from the message composition step for the first
    recipient is used unaltered for all other recipients as well.

2.4. Comparing "sms" URIs

Two "sms" URIs are equivalent according to the following rules.
Since the definition of the <telephone-subscriber> is taken from
[RFC3966], equivalence of individual values of <telephone-subscriber>
is based on the rules defined in Section 4 of [RFC3966], repeated
here for convenience:

o Both must be either a <local-number> or a <global-number<, i.e.,
   start with a "+".

o The <global-number-digits> and the <local-number-digits> must be
   equal, after removing all visual separators.

o For mandatory additional parameters and the <phone-context> and
   <extension> parameters defined in [RFC3966], the <phone-context>
   parameter value is compared either as a host name if it is a
   <domainname> or digit-by-digit if it is <global-number-digits>.
   The latter is compared after removing all <visual-separator>
   characters.

o Parameters are compared according to <pname>, regardless of the
   order they appeared in the URI. If one URI has a parameter name
   not found in the other, the two URIs are not equal.

o URI comparisons are case-insensitive.

Since "sms" URIs can contain multiple <telephone-subscriber>s as well
as <sms-fields>, in addition to adopting the rules defined for
comparing <telephone-subscriber>s as defined by [RFC3966], two "sms"
URIs are only equivalent if their <sms-fields> are identical, and if
all <telephone-subscriber>s, compared pairwise as a set (i.e.,
without taking sequence into consideration), are equivalent.

2.5. Examples of Use

sms:+15105550101

This indicates an SMS-message-capable recipient at the given
telephone number. The message is sent using the user agent's default
SMS delivery method.

sms:+15105550101,+15105550102

This indicates SMS-message-capable recipients at the given telephone
numbers. The identical message should be sent to both recipients
using the user agent's default SMS delivery method.

sms:+15105550101?body=hello%20there

In this case, a message (initially being set to "hello there", which
may be modified by the user before sending) will be sent via SMS
using the user agent's default SMS delivery method.

2.6. Using "sms" URIs in HTML Forms

When using an "sms" type URI as an action URI for HTML form
submission [HTML401], the form contents MUST be packaged in the SMS
message just as they are packaged when using a "mailto" URI
[RFC2368], using the "application/x-www-form-urlencoded" media type
(as defined by HTML [HTML401]), effectively packaging all form data
into URI-compliant syntax [RFC3986]. The SMS message MUST NOT

contain any HTTP header fields, only the form data. The media type
is implicit. It MUST NOT be transferred in the SMS message. Since
the SMS message contains the form field values, the body <sms-field>
of an "sms" type URI used for an HTML form will be ignored.

The character encoding used for form submissions MUST be UTF-8
[RFC3629]. It should be noted, however, that user agents MUST
percent-encode form submissions before sending them (this encoding is
specified by the URI syntax [RFC3986]).

The user agent SHOULD inform the user about the possible security
hazards involved when submitting the form (it is probably being sent
as plain text over an air interface).

If the form submission is longer than the maximum SMS message size,
the user agent MAY either concatenate SMS messages, if it is able to
do so, or it MAY refuse to send the message. The user agent MUST NOT
send out partial form submissions.

3. URI Scheme Registration

This memo requests the registration of the Uniform Resource
Identifier (URI) scheme "sms" for specifying one or more recipients
for an SMS message. The registration request complies with
[RFC4395].

3.1. URI Scheme Name

sms

3.2. Status

Permanent

3.3. URI Scheme Syntax

See Section 2.

3.4. URI Scheme Semantics

The "sms" URI scheme defines a way for a message to be composed and
then transmitted using the SMS message transmission method. This
scheme can thus be compared to be "mailto" URI scheme [RFC2368]. See
Section 2.3 for the details of operation.

3.5. Encoding Considerations

The optional body field of "sms" URIs may contain a message text, but this text uses percent-encoded UTF-8 characters and thus can always be represented using URI characters. See Section 2 for the details of encoding.

3.6. Applications/Protocols That Use This URI Scheme Name

The "sms" URI scheme is intended to be used in a similar way as the "mailto" URI scheme [RFC2368]. By using "sms" URIs, authors can embed information into documents that can be used as a starting point for initiating message composition. Whether the client is sending the message itself (for example, over a GSM air interface) or redirecting the user to a third party for message composition (such as a Web service for sending SMS messages) is outside of the scope of the URI scheme definition.

3.7. Interoperability Considerations

No interoperability issues have been identified.

3.8. Security Considerations

See Section 4.

3.9. Contact

Erik Wilde
School of Information
UC Berkeley
Berkeley, CA 94720-4600
U.S.A.
tel:+1-510-6432252
mailto:dret@berkeley.edu

4. Security Considerations

SMS messages are transported without any provisions for privacy or integrity, so SMS users should be aware of these inherent security problems of SMS messages. Unlike electronic mail, where additional mechanisms exist to layer security features on top of the basic infrastructure, there currently is no such framework for SMS messages.

SMS messages very often are delivered almost instantaneously (if the receiving SMS client is online), but there is no guarantee for when SMS messages will be delivered. In particular, SMS messages between

different network operators sometimes take a long time to be
delivered (hours or even days) or are not delivered at all, so
applications SHOULD NOT make any assumptions about the reliability
and performance of SMS message transmission.

In most networks, sending SMS messages is not a free service.
Therefore, SMS clients MUST make sure that any action that incurs
costs is acknowledged by the end user, unless explicitly instructed
otherwise by the end user. If an SMS client has different ways of
submitting an SMS message (such as a Web service and a phone line),
then the end user MUST have a way to control which way is chosen.

SMS clients often are limited devices (typically mobile phones), and
the sending SMS client SHOULD NOT make any assumptions about the
receiving SMS client supporting any non-standard services, such as
proprietary message concatenation or proprietary content types.
However, if the sending SMS client has prior knowledge about the
receiving SMS client, then he MAY use this knowledge to compose non-
standard SMS messages.

There are certain special SMS messages defined in the SMS
specification [SMS] that can be used, for example, to turn on
indicators on the phone display or to send data to certain
communication ports (comparable to UDP ports) on the device. Certain
proprietary systems (for example, the Wireless Application Protocol
[WAP]) define configuration messages that may be used to reconfigure
the devices remotely. Any SMS client SHOULD make sure that malicious
use of such messages is not possible, for example, by filtering out
certain SMS User Data header fields. Gateways that accept SMS
messages (e.g., in email messages or Web forms) and pass them on to
an SMSC SHOULD implement this kind of "firewalling" approach as well.

Because of the narrow bandwidth of the SMS communications channel,
there should also be checks in place for excessively long
concatenated messages. As an example, it may take two minutes to
transfer thirty concatenated text messages.

Unchecked input from a user MUST NOT be used to populate any other
fields in an SMS message other than the User Data field (not
including the User Data header field). All other parts, including
the User Data header, of the short message should only be generated
by trusted means.

By including "sms" URIs in unsolicited messages (a.k.a. "spam") or
other types of advertising, the originator of the "sms" URIs may
attempt to reveal an individual's phone number and/or to link the
identity (i.e., email address) used for messaging with the identity
(i.e., phone number) used for the mobile phone. This attempt to

collect information may be a privacy issue, and user agents may make
users aware of that risk before composing or sending SMS messages.
Users agents that do not provide any feedback about this privacy
issue make users more vulnerable to this kind of attack.

A user agent SHOULD NOT send out SMS messages without the knowledge
of the user because of associated risks, which include sending masses
of SMS messages to a subscriber without his consent, and the costs
involved in sending an SMS message.

As suggested functionality, the user agent MAY offer a possibility
for the user to filter out those phone numbers that are expressed in
local format, as most premium-rate numbers are expressed in local
format, and because determining the correct local context (and hence
the validity of the number to this specific user) may be very
difficult.

When using "sms" URIs as targets of forms (as described in
Section 2.6), the user agent SHOULD inform the user about the
possible security hazards involved when submitting the form (it is
probably being sent as plain text over an air interface).

5. IANA Considerations

IANA has registered the "sms" URI scheme, using the template in
Section 3, in accordance with [RFC4395].

6. Acknowledgements

This document has been prepared using the IETF document DTD described
in [RFC2629].

Thanks to (listed alphabetically) Claudio Allocchio, Derek Atkins,
Nevil Brownlee, John Cowan, Leslie Daigle, Lisa Dusseault, Miguel
Garcia, Vijay Gurbani, Alfred Hoenes, Cullen Jennings, Graham Klyne,
Larry Masinter, Alexey Melnikov, Michael Patton, Robert Sparks, and
Magnus Westerlund for their comments.

7. References

7.1. Normative References

[Err203]   RFC Errata, "Errata ID 203", RFC 3629,
            <http://www.rfc-editor.org>.

[HTML401]   Raggett, D., Le Hors, A., and I. Jacobs, "HTML 4.01
            Specification", W3C REC-html401, December 1999,
            <http://www.w3.org/TR/html401/>.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3629]   Yergeau, F., "UTF-8, a transformation format of ISO
            10646", STD 63, RFC 3629, November 2003.

[RFC3966]   Schulzrinne, H., "The tel URI for Telephone Numbers",
            RFC 3966, December 2004.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifier (URI): Generic Syntax", STD 66,
            RFC 3986, January 2005.

[RFC4395]   Hansen, T., Hardie, T., and L. Masinter, "Guidelines and
            Registration Procedures for New URI Schemes", BCP 35,
            RFC 4395, February 2006.

[RFC5234]   Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
            Specifications: ABNF", STD 68, RFC 5234, January 2008.

[SMS]   European Telecommunications Standards Institute, "3GPP TS
            23.040 V7.0.1 (2007-03): 3rd Generation Partnership
            Project; Technical Specification Group Core Network and
            Terminals; Technical realization of the Short Message
            Service (SMS) (Release 7)", March 2007, <http://
            www.3gpp.org/ftp/Specs/archive/23_series/23.040/
            23040-701.zip>.

[SMS-CHAR]
            European Telecommunications Standards Institute, "TS 100
            900 (GSM 03.38 version 7.2.0 Release 1998): Digital
            Cellular Telecommunications System (Phase 2+); Alphabets
            and language-specific information", July 1999, <http://
            www.3gpp.org/ftp/Specs/archive/03_series/03.38/
            0338-720.zip>.

7.2. Informative References

[RFC2368]   Hoffmann, P., Masinter, L., and J. Zawinski, "The mailto
            URL scheme", RFC 2368, June 1998.

[RFC2629]   Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,
            June 1999.

[RFC2838]   Zigmond, D. and M. Vickers, "Uniform Resource Identifiers
            for Television Broadcasts", RFC 2838, May 2000.

   [WAP]   WAP Forum, "Wireless Application Protocol - Architecture
               Specification (WAP-210-WAPArch-20010712)", July 2001.

   [uri-clarification]
               World Wide Web Consortium, "URIs, URLs, and URNs:
               Clarifications and Recommendations 1.0", W3C uri-
               clarification , September 2001,
               <http://www.w3.org/TR/uri-clarification/>.

016

Appendix A. Syntax of 'telephone-subscriber'

   The following syntax is reproduced from Section 3 of [RFC3966]. It
   defines the <telephone-subscriber> part used in the "sms" URI scheme
   syntax. Please note that it includes Erratum 203 [Err203] for RFC
   3966, which changes the definition of <isdn-subaddress>.

```
   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*paramchar
   extension            = ";ext=" 1*phonedigit
   context              = ";phone-context=" descriptor
   descriptor           = domainname / global-number-digits
   global-number-digits = "+" *phonedigit DIGIT *phonedigit
   local-number-digits  =
      *phonedigit-hex (HEXDIG / "*" / "#")*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
```

Authors' Addresses

    Erik Wilde
    UC Berkeley
    School of Information
    Berkeley, CA 94720-4600
    U.S.A.

    Phone: +1-510-6432253
    EMail: dret@berkeley.edu
    URI: http://dret.net/netdret/

    Antti Vaha-Sipila
    Nokia

    EMail: antti.vaha-sipila@nokia.com
    URI: http://www.iki.fi/avs/