The Core Session Initiation Protocol User Agent Protocol Data Set
draft-petrie-sipping-sip-dataset-02.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 26, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines the properties and format for the core SIP user agent protocol dataset. The properties defined in this document are expected to be common to most SIP user agents regardless of whether the user agent support audio, video, text or any combination of media. These core SIP properties are considered to be a dataset.
Several datasets may be combined into documents or profiles that are provided to SIP user agents so that they can operate with the desired behavior.

Table of Contents

1. Motivation .............................................. 3
2. Introduction ............................................ 3
   2.1. Requirements Terminology ............................. 3
   2.2. Profile Data Terminology ............................. 3
   2.3. Overview ........................................... 4
3. Core SIP Data Set ....................................... 7
   3.1. Transport Protocol Data Set ......................... 7
      3.1.1. transportProtocols Data Set Properties Definitions .............................. 7
      3.1.2. transportProtocols Element Definition ........................................... 7
      3.1.3. Merging Different Sources of a transportProtocol Data Set ...................... 7
   3.2. outboundProxy ....................................... 8
      3.2.1. outboundProxy Data Set Properties Definitions .................................... 8
      3.2.2. outboundProxies Element Definition ............................................. 8
      3.2.3. outboundProxies Merging Different Sources of a Data Set ......................... 9
   3.3. sipMethods .......................................... 9
      3.3.1. sipMethods Dataset Properties Definitions ..................................... 9
      3.3.2. sipMethod Element Definition ................................................. 9
      3.3.3. sipMethods Merging Different Sources of a Data Set .......................... 9
   3.4. sipOptionTags ....................................... 10
      3.4.1. sipOptionTags Data Set Properties Definitions .................................. 10
      3.4.2. sipOptionTags Element Definition ............................................. 10
      3.4.3. sipOptionTags Merging Different Sources of a Data Set ......................... 10
4. Example Profiles and Use .................................. 10
   4.1. Merge Two Data Sets .................................. 10
   4.2. Policy Filtering ..................................... 12
   4.3. Override ............................................ 12
5. Security Considerations .................................. 12
6. IANA Considerations ..................................... 13
   6.1. Content-type registration for ’application/uapsip+xml’ .......................... 13
7. Change History .......................................... 13
   7.1. Changes from draft-petrie-sipping-sip-dataset-01 ................................ 14
   7.2. Changes from draft-petrie-sipping-sip-dataset-00 ................................ 14
   7.3. Changes from draft-petrie-sipping-profile-datasets-01 ............................ 14
8. Normative References ..................................... 14
   Appendix A. SIP Protocol Dataset Schema .......................... 15
   Appendix B. Acknowledgments ................................. 19
   Authors’ Addresses ......................................... 20
   Intellectual Property and Copyright Statements .................. 21
1. Motivation

The SIP Profile Data Sets defined in this document support the principle to enable SIP User Agents to obtain and use profile data sets from multiple sources in order to support a wide range of applications without undue complexity.

The SIP Protocol Data Set is intended the be the lowest common denominator among all user agent types regardless of capability. This data set contains properties that all user agents require. That does not mean that all of these properties are mandatory.

[I-D.ietf-sipping-config-framework] defines a configuration framework for finding, retrieving and change notification of profile data for SIP [RFC3261] user agents. It is intended that the SIP dataset defined in this document may be contained in the user, device and local-network profiles described in the configuration framework. [I-D.petrie-sipping-profile-datasets] defines a general XML schema to contain user agent profile data. This document defines core SIP specific data by extending the profile data sets schema. The MIME type defined for this schema may be used by the user agent to indicate the support for this schema and content in the profile.

2. Introduction

This document defines the properties and format for the core SIP user agent profile data set. The following properties are defined in this document:

- transportProtocols
- outboundProxies
- sipMethods
- sipOptionTags

and, are expected to be common to most SIP user agents regardless of whether the user agent support audio, video, text or any combination of media. These core SIP properties are considered to to be a data set.

2.1. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in RFC 2119[RFC2119].

2.2. Profile Data Terminology
property - a named configurable characteristic of a user agent. A given property has a well-defined range of possible values. A given property may be defined to have range of values, allow for simultaneous use of many values (as in a list of allowed possibilities), or be a set of related values that collectively form a single profile information item.

setting - the binding of a specific value or set of values to a given property.

profile - a collection of settings to be applied for a specific user, device, or local network.

device - SIP user agent, either software or hardware appliance. This is a logical concept, as there may be no physical dedicated device or it may be part of an assembly of devices. In this document, the terms "user agent" and "device" are interchangeable.

user profile - the profile that applies to a specific user. This is best illustrated by the "hotelling" use case - a user has an association for some period of time with a particular device. The user profile is that set of profile data the user wants to associate with that device (e.g. ringtones used when someone calls them, the user’s shortcuts).

device profile - data profile that applies to a specific device. In the "hotelling" use case, this is the data that is bound to the device itself independent of the user. It relates to specific capabilities of the device and/or preferences of the owner of the device.

local network profile - data that applies to the user agent in the context of the local network. This is best illustrated by roaming applications; a new device appears in the local network (or a device appears in a new network, depending on the point of view). The local network profile includes settings and perhaps policies that allow the user agent to function in the local network (e.g. how to traverse NAT or firewall, bandwidth constraints).

data set - a collection of properties.

working profile - the set of property values actually set in a SIP User Agent as a result of merging the profiles from all sources; the actual effective profile for the user agent.

merging - the operation of resolving overlapping settings from multiple profiles. Overlap occurs when the same property occurs in multiple profiles (e.g. user, device, local network).

2.3. Overview

This document defines a new MIME type: application/uapsip+xml which SHOULD be used by user agents to indicate support for this schema. The XML namespace: urn:ietf:params:xml:ns:uaprof:sip is used for the extensions to the SIP UA Profile Dataset [I-D.petrie-sipping-profile-datasets] schema defined here.
The Core SIP UA Protocol dataset is defined in Section 3 and complies with the guidelines provided in Section 5 of [I-D.petrie-sipping-profile-datasets]. The Relax NG Schema defined in this document extends the profile datasets schema to define the core SIP settings which are likely to be common to all types of SIP user agents. The SIP transport protocol(s) and ports, outbound proxy(s), SIP methods and option tags to be enabled on the user agent may be configured with this dataset.

Section 4 provides illustrative example profiles and use cases for merging. Security considerations are addressed in Section 5.

The following is an example instance of the SIP protocol data set. Note the use of the policy attribute.
<?xml version="1.0" encoding="UTF-8"?>
<propertySet xmlns="urn:ietf:params:xml:ns:uaprof">
  <profileUri>sip:a1b2c3d4e5f6@example.com</profileUri>
  <profileCredential>
    <realm>example.com</realm>
    <authUser>fred</authUser>
    <a1Digest>b6b577fd12aa7e1df8d60735ef56fc2e</a1Digest>
    <!-- <password>123</password> -->
  </profileCredential>
  <profileContactUri>tel:+16175551212</profileContactUri>
  <profileContactUri>sip:411@example.com</profileContactUri>
  <profileContactUri>http://example.com/sipProfile.html</profileContactUri>
  <profileInfo>
    This is an example profile from example.com
  </profileInfo>
    <transportProtocol policy="allow">
      <name>UDP</name>
      <port>5092</port>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>TCP</name>
      <port>5092</port>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>TLS</name>
      <port>5092</port>
    </transportProtocol>
  </transportProtocols>
  <outboundProxies xmlns="urn:ietf:params:xml:ns:uaprof:sip">
    <outboundProxy q="0.1">sip:example.com</outboundProxy>
    <outboundProxy q="0.2">sip:dmz.example.com</outboundProxy>
    <outboundProxy q="0.3">sip:local.example.com</outboundProxy>
  </outboundProxies>
  <sipMethods xmlns="urn:ietf:params:xml:ns:uaprof:sip">
    <sipMethod policy="disallow">REFER</sipMethod>
    <sipMethod policy="allow">METHOD</sipMethod>
  </sipMethods>
  <sipOptionTags xmlns="urn:ietf:params:xml:ns:uaprof:sip">
    <sipOptionTag policy="allow">session</sipOptionTag>
    <sipOptionTag policy="allow">join</sipOptionTag>
  </sipOptionTags>
</propertySet>
3. Core SIP Data Set

The XML schema defined in this document extends the root element "propertySet" schema defined in I-D.petrie-sipping-profile-datasets.

3.1. Transport Protocol Data Set

3.1.1. transportProtocols Data Set Properties Definitions

TransportProtocols - This property contains properties related to SIP transport protocols, and is an XML element that extends on the XML "setting_container" element contained in the root "propertySet" element. It serves as a container for a list of SIP transport protocols to allow or disallow. There may be zero or one elements.

3.1.2. transportProtocols Element Definition

TransportProtocol - The "transportProtocol" is an XML element that extends the "setting" element contained in the "transportProtocols" element. The "transportProtocol" element contains properties related to a SIP transport protocol. It names the transport protocol, defines whether the protocol is enabled or not and defines the port to which that protocol is bound. If the protocol is named it defaults to enabled if not explicitly set. If the port property is not set, it defaults to the default specified by the specification which binds the protocol to SIP. The user agent should enable all the set transport protocols that are supported by the user agent. The user agent ignores protocol bindings that it does not support. The user agent may default transport protocols that it supports to enabled, if a protocol property for that transport protocol is not present in the data set. The order of the list of transportProtocol setting values indicated by the "q" attribute indicates the order of preference. There may be zero or more "transportProtocol" elements in the "transportProtocols" element.

Name - This XML element identifies the specific transport protocol, and extends the "setting" element contained in transportProtocols. There must be exactly one "name" element in a "transportProtocol".

Port - This element identifies the port for binding the transport protocol, and extends the "setting" element contained in transportProtocols. There must be exactly one "port" element in a "transportProtocol".

3.1.3. Merging Different Sources of a transportProtocol Data Set

The "transportProtocol" property uses the "policy" attribute to identify whether the transport protocol is allowed or disallowed. The "q" attribute is used for ordering of the list. In addition, a
visibility attribute may be present.

If there are matches on multiple "name" element values, the "policy" attribute will determine which is allowed or not. As defined in Merging Datasets [I-D.petrie-sipping-profile-datasets] properties with conflicting "policy" attribute value of "allow" and "disallow" are assumed to be "disallowed". If there are multiple "transportProtocol" elements from different profiles with the same "name" element value and "policy" attribute values of "allows", then the resulting merged "transportProtocols" element will contain one "transportProtocol" element having a "name" element of that value. The "port" element value will be determined in the following order of the source profile, when there are multiple "transportProtocol" elements from different profiles with the same "name" element value and "policy" attribute value of "allow":

- Local
- Device
- User
- Application

3.2. outboundProxy

3.2.1. outboundProxy Data Set Properties Definitions

outboundProxies - The "outboundProxies" property is an XML element that extends on the XML "setting" element contained in the root "propertySet" element. It serves as a container for a list of outbound proxies. There may be zero or one element. The default outbound proxy, through which all SIP requests, not explicitly routed, should be sent. The format of this parameter is of name-addr as specified in [RFC3261]. This property is optional. If absent or not set, SIP requests are sent to directly to the URI of the request. If set the effect of this property is to add a loose route as defined in [RFC3261] for the next hop destination.

3.2.2. outboundProxies Element Definition

outboundProxies - The "outboundProxy" is an XML element that extends the XML "setting" element contained in "outboundProxies". There may be zero, one or many "outboundProxy" elements. It provides default value for an outbound proxy, through which all SIP requests, not explicitly routed, should be sent. The format of this parameter is of name-addr as specified in [RFC3261]. This property is optional. If absent or not set, SIP requests are sent to directly to the URI of the request. If set the effect of this property is to add a loose route as defined in [RFC3261] for the next hop destination. Multiple "outboundProxy" elements may be contained in the "outboundProxies" element to form a route set.
The user agent MUST use this route set for all requests that are outside of or initiate a dialog. The user agent MUST NOT use this route set for requests within an established dialog. The user agent should use the route set formed as described in [RFC3261] for requests within an established dialog.

3.2.3.  outboundProxies Merging Different Sources of a Data Set

The aggregation approach is used to resolve conflicts. By aggregating the multiple outbound proxies, the local network profile provided outbound proxy allows the signaling to get out of the local network and the device profile provided outbound proxy is able to monitor all SIP signaling from the user agent. The order of the resulting merged, route set is determined by the "q" attribute.

3.3.  sipMethods

3.3.1.  sipMethods Dataset Properties Definitions

sipMethods - This property contains properties related to SIP Methods, and is an XML element that extends on the XML "setting" element contained in the root "propertySet" element. It serves as a container for a list of SIP request methods to allow or disallow. Typically, only provide by the device dataset. The "sipMethods" element is intended to provide a means of enabling or disabling features in the SIP user agent based upon the SIP request method.

3.3.2.  sipMethod Element Definition

sipMethod - An element to specify a SIP method, and extends the "setting" element contained in the "sipMethods" element. There may be zero or more elements. For user agents that support the method indicated, this element serves as a switch to enable or disable the named SIP method as indicated by the "policy" attribute. The direction attribute is used to indicate asymmetric support of the method. The sendonly value in the direction attribute means that the user agent MAY send requests of the give method name but MUST reject all incoming requests for the method type.

3.3.3.  sipMethods Merging Different Sources of a Data Set

The "sipMethods" Data Set uses the aggregation merging policy defined in [I-D.petrie-sipping-profile-datasets]. When multiple "sipMethod" elements with the same value are provided, the "policy" attribute is used to determine precedence.
3.4. sipOptionTags

3.4.1. sipOptionTags Data Set Properties Definitions

sipOptionTags - This property specifies a container for a list of SIP option tags that are allowed or disallowed, and is an XML element that extends on the XML "setting" element contained in the root "propertySet" element. For user agents that support features indicated by option tags, this element serves as a list of features to turn on or off as indicated by the "policy" attribute in the "sipOptionTag" element.

3.4.2. sipOptionTags Element Definition

sipOptionTag - An element to specify a SIP option tag, and extends the "setting" element and is contained in "sipOptionTags" element. There may be zero or more elements "sipOptionTag". For user agents that support features indicated by option tags, this element serves as a switch to enable or disable the named SIP option as indicated by the policy attribute in the "sipOptionTag" element.

3.4.3. sipOptionTags Merging Different Sources of a Data Set

The sipOptionTags Data Set uses the default aggregation merging policy defined in [I-D.petrie-sipping-profile-datasets]. When multiple "sipMethod" elements with the same value are provided, the "policy" attribute is used to determine precidence.

4. Example Profiles and Use

4.1. Merge Two Data Sets

Consider the use case described in [I-D.petrie-sipping-profile-datasets] where the user wishes to indicate that only secure SIP transport should be used. The device profile may contain SIP Protocol Data Set (see Section 3.1) settings that look like the following:
The user profile which indicates that only TLS should be used would look like (Note: this example also indicates that port 5061 should be used. This may be more constrained than the user really wants.):

```
<propertySet>
  <transportProtocols>
    <transportProtocol policy="allow">
      <name>TLS</name>
      <port>5061</port>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>UDP</name>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>TCP</name>
    </transportProtocol>
  </transportProtocols>
  <outboundProxies>
    <outboundProxy>
      sip:outproxy.example.com
    </outboundProxy>
  </outboundProxies>
  <sipMethods>
    <sipMethod policy="disallow">INFO</sipMethod>
  </sipMethods>
  <sipOptionTags>
    <sipOptionTag policy="disallow">join</sipOptionTag>
  </sipOptionTags>
</propertySet>
```
The merged result of the device and user profile would look like:

```xml
<propertySet>
  <transportProtocols>
    <transportProtocol policy="allow">
      <name>TLS</name>
      <port>5061</port>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>UDP</name>
    </transportProtocol>
    <transportProtocol policy="disallow">
      <name>TCP</name>
    </transportProtocol>
  </transportProtocols>
  <outboundProxies>
    <outboundProxy>
      sip:outproxy.example.com
    </outboundProxy>
  </outboundProxies>
  <sipMethods>
    <sipMethod policy="disallow">INFO</sipMethod>
  </sipMethods>
  <sipOptionTags>
    <sipOptionTag policy="disallow">join</sipOptionTag>
  </sipOptionTags>
</propertySet>
```

4.2. Policy Filtering

(allowed and disallowed protocols)

4.3. Override

(device prefers default ports 5060, local net requires port 11000)

5. Security Considerations

Security is mostly a profile delivery problem. The delivery framework MUST provide a secure means of delivering the profile data as it may contain sensitive data that would be undesirable if it were stolen or sniffed. Storage of the profile on the profile delivery server and user agent is an implementation problem. The profile delivery server and the user agent MUST provide protection that prevents unauthorized access of the profile data. The profile
delivery server and the user agent MUST enforce the access control policies defined in the profile data sets if present.

6. IANA Considerations

XML name space registration: urn:ietf:params:xml:ns:uaprof:sip

6.1. Content-type registration for 'application/uapsip+xml'

To: ietf-types@iana.org
Subject: Registration of MIME media type application/uapsip+xml
MIME media type name: application
MIME subtype name: uapsip+xml
Required parameters: (none)
Optional parameters: charset Indicates the character encoding of enclosed XML. Default is UTF-8.
Encoding considerations: Uses XML, which can employ 8-bit characters, depending on the character encoding used. See RFC 3023 [RFC 3023], section 3.2.
Security considerations: This content type is designed to carry SIP user agent profile data, which may be considered private information. Appropriate precautions should be adopted to limit disclosure of this information.
Interoperability considerations: This content type provides a common format for exchange of SIP user agent profile SIP settings information.
Published specification: RFC XXXX (Note to RFC Editor: Please fill in XXXX with the RFC number of this specification)
Applications which use this media type: SIP user agents and profile delivery servers.
Additional information: Magic number(s): File extension(s): Macintosh File Type Code(s):
Person & email address to contact for further information: Daniel Petrie EMail: dan.ietf AT sipez DOT com
Intended usage: LIMITED USE
Author/Change controller: This specification is a proposed work item of the IETF SIPPING working group, with mailing list address: sipping@ietf.edu.
Other information: This media type is a specialization of application/xml [RFC 3023], and many of the considerations described there also apply to application/uapsip+xml.

7. Change History

[[RFC Editor: Please remove this entire section upon publication as an RFC.]]
7.1. Changes from draft-petrie-sipping-sip-dataset-01

Converted the XML schema to use Relax NG and created a valid schema.

Defined XML name space for schema:
"urn:ietf:params:xml:ns:uaprof:sip"

Changed names of elements, attributes and other data types which contained "-" or "_" to use camel case.

Defined mime type: application/uapsip+xml to be used for indication of support for this content type (e.g. with the config. framework).

7.2. Changes from draft-petrie-sipping-sip-dataset-00

Removed references to policy attribribute value: "mandatory".

A few minor edits to refresh the draft

7.3. Changes from draft-petrie-sipping-profile-datasets-01

The core SIP profile data set was split out from the examples in draft-petrie-sipping-profile-datasets-01 to create a stand alone data set definition.

8. Normative References

[I-D.ietf-sipping-config-framework]
Petrie, D., "A Framework for Session Initiation Protocol User Agent Profile Delivery",
draft-ietf-sipping-config-framework-09 (work in progress), October 2006.

[I-D.petrie-sipping-profile-datasets]
Petrie, D., "A Schema and Guidelines for Defining Session Initiation Protocol User Agent Profile Data Sets",
draft-petrie-sipping-profile-datasets-03 (work in progress), October 2005.


[W3C.REC-xml-names]

[W3C.REC-xmlschema-1]

[W3C.REC-xmlschema-2]

Appendix A. SIP Protocol Dataset Schema

The following is the schema for the SIP protocol data set.

```xml
<?xml version="1.0"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
          ns="urn:ietf:params:xml:ns:uaprof:sip"
          datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
    <include href="uaprofile.rng">
        <define name="ElementGenericSetting">
            <ref name="ElementGenericSipSetting"/>
        </define>
        <define name="ElementGenericSettingContainer">
            <ref name="ElementGenericSipSettingContainer"/>
        </define>
    </include>
    <define name="ElementGenericSipSettingContainer">
        <element>
            <anyName>
                <except>
                    <nsName ns="urn:ietf:params:xml:ns:uaprof"/>
                    <nsName ns="urn:ietf:params:xml:ns:uaprof:sip"/>
                    <nsName ns=""/>
                </except>
            </anyName>
        </ref>
        <ref name="SettingContainerAttributes"/>
    </element>
</grammar>
```

<!-- container can have containers or settings not both -->
<choice>
  <zeroOrMore>
    <ref name="ElementGenericSipSetting"/>
  </zeroOrMore>
  <zeroOrMore>
    <ref name="ElementGenericSipSettingContainer"/>
  </zeroOrMore>
</choice>
</define>

<define name="ElementGenericSipSetting">
  <element>
    <anyName>
      <except>
        <nsName ns="urn:ietf:params:xml:ns:uaprof"/>
        <nsName ns="urn:ietf:params:xml:ns:uaprof:sip"/>
        <nsName ns=""/>
      </except>
    </anyName>
    <ref name="SettingAttributes"/>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <choice>
      <text/>
      <ref name="ElementGenericSipSetting"/>
    </choice>
    <zeroOrMore>
  </element>
</define>

<!-- Extend elements in root/propertySet -->
<define name="PropertySetExtension" combine="interleave">
  <optional>
    <ref name="ElementTransportProtocols"/>
  </optional>
  <optional>
    <ref name="ElementOutboundProxies"/>
  </optional>
  <optional>
    <ref name="ElementSipMethods"/>
  </optional>
  <optional>
    <ref name="ElementSipOptionTags"/>
  </optional>
</define>
<define name="ElementTransportProtocols">
  <element name="transportProtocols">
    <ref name="SettingContainerAttributes"/>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="ElementTransportProtocol"/>
    </zeroOrMore>
  </element>
</define>

<define name="ElementTransportProtocol">
  <element name="transportProtocol">
    <optional>
      <ref name="AttributePolicy"/>
    </optional>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <element name="name">
      <ref name="DataIpTransport"/>
    </element>
    <element name="port">
      <ref name="DataIpPort"/>
    </element>
  </element>
</define>

<define name="ElementOutboundProxies">
  <element name="outboundProxies">
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="ElementOutboundProxy"/>
    </zeroOrMore>
  </element>
</define>

<define name="ElementOutboundProxy">
  <element name="outboundProxy">
    <optional>
      <ref name="AttributeQ"/>
    </optional>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <ref name="DataSipUri"/>
  </element>
</define>
<define name="ElementSipMethods">
  <element name="sipMethods">
    <ref name="SettingContainerAttributes"/>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="ElementSipMethod"/>
    </zeroOrMore>
  </element>
</define>

<define name="ElementSipMethod">
  <element name="sipMethod">
    <optional>
      <ref name="AttributePolicy"/>  
    </optional>
    <optional>
      <ref name="AttributeDirection"/>
    </optional>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <data type="string">
      <param name="pattern">[A-Z]*</param>
    </data>
  </element>
</define>

<define name="ElementSipOptionTags">
  <element name="sipOptionTags">
    <ref name="SettingContainerAttributes"/>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="ElementSipOptionTag"/>
    </zeroOrMore>
  </element>
</define>

<define name="ElementSipOptionTag">
  <element name="sipOptionTag">
    <optional>
      <ref name="AttributePolicy"/>
    </optional>
    <optional>
      <ref name="AttributeDirection"/>
    </optional>
    <zeroOrMore>
      <ref name="AttributeGeneric"/>
    </zeroOrMore>
  </element>
</define>
Appendix B. Acknowledgments
Authors’ Addresses

Daniel Petrie
SIPez LLC.
34 Robbins Rd.
Arlington, MA  02476
US

Phone: +1 617 273 4000
Email: dan.ietf AT SIPez DOT com
URI:  http://www.sipez.com/

Martin Dolly
AT&T Labs
200 Laurel Avenue
Middletown, NJ  07748
US

Phone:  
Email: mdolly AT att DOT com
URI:  

Volker Hilt
Bell Labs/Lucent Technologies
101 Crawfords Corner Rd
Holmdel, NJ  07733
US

Phone:  
Email: volkerh@bell-labs.com
URI:  
