

User Agent Configuration Recommendation

DRAFT

Copyright © 2010 [SIP Forum](#) (a Swedish non-profit association), and SIP Forum LLC. All rights reserved.

DRAFT

COLLABORATORS

	<i>TITLE :</i> User Agent Configuration Recommendation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Scott Lawrence, Avaya Inc. - Editor John Elwell, Siemens Enterprise Communications - Task Group Chair	March 2, 2010	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
0.24		<p>This revision moves the definition and use of the Provider Alias Number resolution mechanism to the non-normative Appendix B, Numeric Identifiers for SIP Service Providers.</p> <p>This document depends on three Internet Drafts that are in various stages of review in the IETF (see Internet Draft References). There are links in the reference entries to tracking data at the IETF to reflect the current status of those drafts. It is possible that changes made to one or more of those documents before they are issued as RFCs will require corresponding changes in this document; the Task Group believes that any such changes will be minor. Aside from updating those references, this specification is now stable - this revision is the candidate text for the published version 1.0 of this Recommendation.</p>	
0.23	2010-02-12		
0.22	2010-01-27		
0.21	2010-01-22		
0.20	2009-12-01		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
0.19	2009-11-17		
0.18	2009-11-12		
0.17	2009-11-04		
0.16	2009-10-26		
0.15	2009-10-19		
0.14	2009-10-07		
0.13	2009-09-23		
0.12	2009-08-21		
0.11	2009-08-02		
0.10	2009-07-02		
0.9	2009-06-29		
0.8	2009-06-04		
0.7	2009-05-18		
0.6	2009-04-13		
0.5	2009-03-18		
0.4	2009-02-27		
0.3	2009-02-19		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
0.2	2009-01-29		
0.1	2009-01-22		
0.0	2008-12-18		

DRAFT

Contents

1	Introduction	1
1.1	Scope	1
1.2	Terminology	1
1.3	User Agent Installation Examples	2
1.3.1	Hosted IP Service Provider Example	2
1.3.2	IP-PBX Example	2
1.3.3	Special Considerations for High Security Deployments	2
2	Obtaining User Agent Configuration	3
2.1	Network Discovery	3
2.1.1	Link Layer Provisioning	3
2.1.2	Network Layer Provisioning	3
2.2	Obtaining the Configuration Service Domain	3
2.2.1	The Local Network Domain	4
2.2.2	Manual Domain Name Entry	4
2.3	Constructing the Configuration Request URL	4
2.3.1	Obtaining a Configuration Service Base URL	4
2.3.1.1	Configuration Service Redundancy	4
2.3.1.2	Configuration Service Name to Base URL Resolution Failure	5
2.3.2	Adding Configuration Request Parameters	5
2.3.3	Configuration Request URI Example	6
2.4	Obtaining Configuration from the Configuration Service	6
2.4.1	Configuration Data Request Authentication	6
2.4.2	Configuration Data Request Failure	7
2.5	Requesting Configuration Change Notification	7
2.5.1	Configuration Change Subscriptions	7
2.5.1.1	Change Subscription Failure	8
2.5.1.2	Change Subscription Termination	8
2.5.2	Configuration Change Notices	8
2.6	Validity of Stored Configuration Data	9
2.6.1	Re-validating Configuration Data	9
2.7	Retry Backoff Procedure	9
3	Configuration Data	10
3.1	Configuration Data Items	10
3.1.1	Address-of-Record	10
3.1.2	Realm	10
3.1.3	Username	10
3.1.4	Digest	10
3.1.5	OutboundProxy	10
3.2	Reset User Agent to Default Configuration	10

4	Internet Draft References	10
5	Referenced Standards	11
A	Contributing Members of the SIP Forum User Agent Configuration Working Group	12
B	Numeric Identifiers for SIP Service Providers	12
B.1	Numeric Domain Alias Entry	12
B.1.1	Provider Alias Number Resolution Example	13

DRAFT

Abstract

This document defines procedures for how a SIP User Agent should locate, retrieve, and maintain current configuration information from a Configuration Service.

This document is the product of the SIP Forum Technical Working Group User Agent Configuration Task Group.

DRAFT

1 Introduction

A user gets a new SIP User Agent (UA); it may be a hardware device or software. Some User Agents have a user interface that can accept a username, password, and domain name. Other devices, like Analog Telephony Adapters (ATAs), have no user interface other than that provided by an attached analog phone. How does a non-technical user minimally configure it so that when it is started, something useful happens?

1.1 Scope

This SIP Forum User Agent Configuration Recommendation specifies a procedure for how a SIP User Agent locates, retrieves, and maintains current configuration information for a given SIP Service Provider. As such, it specifies requirements to be met by both the User Agent, the Configuration Service at the SIP Service Provider, and the network infrastructure services that allow them to communicate.

Nothing in this Recommendation prohibits a User Agent from obtaining configuration information by any means in addition to the mechanisms specified by this Recommendation.

The intent of this Recommendation is to provide mechanisms sufficient for User Agents to discover an appropriate source of configuration and maintain the currency of that configuration. A User Agent implementation compliant to this Recommendation is free to also implement additional mechanisms that may be required in particular environments, or for use when the services specified here are not available.

The form and content of configuration data to be downloaded are outside the scope of this Recommendation, although Section 3.1, “**Configuration Data Items**” suggests a minimum set of data items likely to be required by all types of UA.

1.2 Terminology

The following terms are used in this document:

User Agent, UA

As defined in [RFC 3261](#). Note that this includes any implementation of a User Agent. A SIP phone is a User Agent, but the term also encompasses any other entity that uses SIP (for example: for a text chat, for sharing a whiteboard or for fax).

Soft User Agent, Soft UA

A User Agent that runs as an application within some larger system that has responsibility for some of the steps described in this Recommendation. In those cases the Soft UA must be able to obtain the information from the platform. In all cases, the term User Agent also encompasses a Soft User Agent.

SIP Service Provider, Service Provider

An entity that provides services to User Agents using the SIP protocol. This Recommendation requires that a Service Provider make configuration data and certain other information available in order to configure User Agents.

Configuration

The set of information that establishes operational parameters for a particular User Agent.

Configuration Service (CS)

The source of Configuration for User Agents.

Configuration Service Domain

The DNS name for the service from which a Configuration is requested.

MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, OPTIONAL

When used in all capital letters as here, are to be interpreted as described in [\[RFC2119\]](#) *Key words for use in RFCs to Indicate Requirement Levels* .

1.3 User Agent Installation Examples

This section is non-normative; it is a set of "user stories" - narrative descriptions of the user experience in different environments. These are "black box" descriptions meant to include the actions to be taken by the human participants (including administrators and system operators as well as the "user" of the UA), but not how the network elements communicate or operate internally. The intent is that these narratives provide context for the subsequent technical specifications.

1.3.1 Hosted IP Service Provider Example

Configuring a new UA to use a hosted IP telephony service will typically proceed as follows: The customer makes a request to their Service Provider to add one or more new users to their service. The customer may supply further details such as a preferred username, type of end-point and any requests for specific functionality, depending on what information the Service Provider considers useful, but no additional information is required from the customer.

The Service Provider performs any necessary provisioning actions on their equipment, and returns to the customer provisioning information, which may include: a domain name or a numeric domain identifier for the provider, a user identifier, and a password. Typically, a Service Provider will supply provisioning information for each device to be provisioned, but may choose to supply information that can be used with multiple devices, or for a limited duration or with other benefits and restrictions.

The customer enters the provisioning information into the UA to be configured, whereupon the UA uses this information to locate the configuration service, securely fetch the configuration information, and configure itself for operation.

1.3.2 IP-PBX Example

Configuring a new UA in a typical business begins by provisioning a user identity in the PBX (add user "John Smith"), and assigning a phone number to the user. That number must then be assigned to a line on a specific UA; this is usually done by selecting a UA and provisioning it in the PBX by its serial number (usually a MAC address), and then assigning the identity or phone number to a 'line' on that UA in the PBX configuration system.

Once provisioning in the PBX is complete, the new user goes to his or her workplace and connects the UA to the network. When connected and powered up, the UA is provided with the user identity, phone number, and any other configuration data with no local user interaction - just connecting it to the network loads the configuration from the PBX and the UA is operational.

1.3.3 Special Considerations for High Security Deployments

To deploy a new UA in a high security scenario requires some special consideration. A security conscious deployment will most likely require that the SIP and other management interfaces, including the interface to the configuration service, are secured before the device is put in to service.

In order to achieve any level of security, the device will need to be pre-configured with some security related information in the form of certificates. This may be achieved in a number of ways. Some examples include

1. An administrator who configures the device in a secure environment before making the device available to the user.
2. Some certificates may be built into the device during the manufacturing process enabling the configuration service to certify information such as the manufacturer, UA type and MAC address. The configuration service may then be used to provision the device with other certificates as required.
3. The device may have a facility for the user to provide the security information in the form of a security card or dongle.

All these mechanism are likely to restrict the user to a limited set of devices approved for use in a particular deployment.

2 Obtaining User Agent Configuration

This section specifies how a User Agent connects to the network, determines what domain to request configuration for, obtains configuration from that domain, and is notified by that domain when the configuration changes.

The User Agent MAY obtain configuration information by any means in addition to those specified here, and MAY use such information in preference to any of the steps specified below, but MUST be capable of using these procedures alone in order to be compliant with this Recommendation.

2.1 Network Discovery

A UA needs a minimum set of parameters to allow it to communicate on the network. Some networks allow the UA to automatically discover these parameters, while other networks require some or all of these parameters to be manually provisioned on the UA.

2.1.1 Link Layer Provisioning

The UA SHOULD attempt to use LLDP-MED (see [ANSI.TIA-1057-2006]) for automatic provisioning of link layer parameters. In some deployments, failure to properly provision the link layer may result in the UA having incorrect layer 2 priority, degrading the quality of service, or being on the wrong virtual LAN (VLAN), possibly resulting in complete loss of service.

2.1.2 Network Layer Provisioning

In order to communicate using IP, the UA needs the following minimal IP configuration parameters:

IP NETWORK PARAMETERS

- UA IP Address
- Subnet Mask
- Gateway IP address
- DNS Server IP address(es)

With the exception of a Soft UA that relies on its platform to obtain the IP Network Parameters,

- If the User Agent is using IP version 4 on a network technology for which the Dynamic Host Configuration Protocol (DHCP) [RFC2131] is defined, the UA MUST attempt to obtain the IP Network Parameters using DHCP and MUST request DHCP option 15 [RFC2132] for obtaining a domain name.
- If the User Agent is using IP version 6 on a network technology for which the Dynamic Host Configuration Protocol version 6 (DHCPv6) [RFC3315] is defined, the UA MAY use any standard IPv6 mechanism to determine the IP Network Parameters, but MUST request DHCPv6 option 21 [RFC3319] to obtain one or more domain names.

In either case, if the DHCP or DHCPv6 service provides a domain name value or values for the option concerned, the UA MUST save those domain names as candidates for use as the Local Network Domain (see Section 2.2, “Obtaining the Configuration Service Domain”).

2.2 Obtaining the Configuration Service Domain

To obtain a configuration, the UA needs to know what domain to request it from. This domain is the *Configuration Service Domain*; its value is a DNS name (see [RFC1034]).

User control or prior configuration MAY establish a value for the Configuration Service Domain that takes precedence over the discovery procedure defined below. In the absence of user control or prior configuration, the Configuration Service Domain is obtained as specified in Section 2.2.1, “The Local Network Domain”, or if that is unsuccessful, by the manual mechanism specified in Section 2.2.2, “Manual Domain Name Entry”.

2.2.1 The Local Network Domain

The UA MUST attempt to use the Local Network Domain name (see Section 2.1.2, “[Network Layer Provisioning](#)”) as the Configuration Service Domain. If multiple DNS names are provided by DHCPv6 option 21, the UA MUST attempt to use each of the names provided, in the order they were given by the DHCPv6 service, until a configuration is successfully obtained.

If the DHCP service does not provide any local domain name value, the UA SHOULD use the manual mechanism defined in Section 2.2.2, “[Manual Domain Name Entry](#)”.

2.2.2 Manual Domain Name Entry

A UA MAY provide an interface by which the DNS name value of the Configuration Service Name is provided directly by the user.

2.3 Constructing the Configuration Request URL

Using the Configuration Service Domain name obtained in Section 2.2, “[Obtaining the Configuration Service Domain](#)”, the UA MUST construct an HTTPS URL with which to request configuration. Constructing this URL consists of two parts:

- Section 2.3.1, “[Obtaining a Configuration Service Base URL](#)”
- Section 2.3.2, “[Adding Configuration Request Parameters](#)”

2.3.1 Obtaining a Configuration Service Base URL

The Configuration Service Domain is resolved to one or more URLs using the U-NAPTR DDDS application defined in [\[RFC4848\] Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service \(DDDS\)](#).

The lookup key for the U-NAPTR request is the Configuration Service Domain Name determined in Section 2.2, “[Obtaining the Configuration Service Domain](#)”. The UA MUST make a DNS request for NAPTR records for that domain name. From the returned records, the UA MUST select those whose Service field value is "SFUA.CFG"; from those records, the UA MUST extract the https URL of the Configuration Service from the Regular Expression field (see next paragraph for the construction of that field value).

The NAPTR records for the Configuration Service Domain Name whose Service field value is "SFUA.CFG" MUST be configured with the Flag field set to "U", an empty Substitution field, and a Regular Expression field value of the following syntax (i.e., a regular expression to replace the domain name with an https URI):

```
u-naptr-regexp = "!.^.*!URI!"
```

where *URI* is as defined in STD 66 [\[RFC3986\]](#), the URI syntax specification, and where the scheme of the URI is "https".

Note that the UA does not need to implement a general regular expression evaluator in order to process the record above correctly. The URI value can be extracted by stripping the fixed value "!.^.*!" from the beginning of the value, and "!" from the end of the value to obtain the base URL. See Section 2.3.3, “[Configuration Request URI Example](#)”.

2.3.1.1 Configuration Service Redundancy

Multiple Configuration Servers can be used to provide redundancy and additional capacity for provisioning User Agents. If the DNS NAPTR request for the Configuration Service Domain Name returns multiple records with the 'SFUA.CFG' service tag, then the UA should treat the resulting URLs as alternatives, ordered according to the rules for the priority and weight as specified for NAPTR records.

In addition to redundancy provided by multiple NAPTR records, resolution of the host part of the https URL can produce multiple results when resolving the host name.

2.3.1.2 Configuration Service Name to Base URL Resolution Failure

If the DNS request to resolve the Configuration Service Name to a request URL does not receive any response, the UA should follow standard DNS retry procedures.

If the DNS request to resolve the Configuration Domain Name to a host name returns a response that indicates that no matching result is available (NXDOMAIN), the UA SHOULD attempt to obtain another Configuration Domain Name using the procedures in Section 2.2, “[Obtaining the Configuration Service Domain](#)”.

2.3.2 Adding Configuration Request Parameters

To construct the full configuration request URL, the UA adds one or more parameters to the base URLs to specify what configuration the UA is requesting.

1. The UA MUST add all parameters from those defined in the [Configuration Request Parameters](#) list below for which the UA has a value. Any parameter from that set for which the UA does not have a value MUST be omitted.
2. The query parameter names defined by this Recommendation all begin with the prefix 'sfua-'. All names beginning with the prefix 'sfua-' are reserved to for this Recommendation and future revisions. The UA MUST NOT include any request parameter whose name begins with the prefix 'sfua-' that is not defined by this Recommendation (including any future revisions).
3. Any parameter not defined by the Recommendation is allowed, but MAY be ignored by any Configuration Service that does not recognize it.

CONFIGURATION REQUEST PARAMETERS

sfua-id

The URN identifying the User Agent, constructed as specified in section 4.1 of [[RFC5626](#)] *Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)*.

Since the procedure defined by [RFC 5626](#) allows any UA to construct a value for this parameter, the sfua-id parameter MUST always be included.

If the UA implements [RFC 5626](#), and includes the '+sip.instance' Contact header field parameter in any request, when requesting configuration it MUST use the same value for the sfua-id parameter.

sfua-user

An identifier for a user associated with the configuration. Note that this might be different than any SIP 'user' in the UA configuration: it could, for example, be the login name of an account on the service provider web site. The syntax of this parameter is that of the [RFC 2617](#) 'userid'.

See Section 2.4.1, “[Configuration Data Request Authentication](#)” for how this parameter relates to authentication of the configuration data request.

sfua-vendor

An identifier that specifies the vendor of the User Agent. The syntax of the value of this parameter is that of a DNS domain. The domain value MUST be that of a domain owned by the vendor.

sfua-model

An identifier that further specifies the User Agent from among those produced by the vendor. The syntax of the value of this parameter is the same as the [RFC 3261](#) 'token'. Values for this parameter are selected by the vendor.

sfua-revision

An identifier that further specifies the User Agent from among those produced by the vendor. The syntax of the value of this parameter is the same as the [RFC 3261](#) 'token'. Values for this parameter are selected by the vendor.

Security Note: The values of some or all of the request parameters may be sensitive information. Since the configuration data request is made over a TLS connection, the confidentiality of that information is protected on the network. Configuration Service implementations should take all necessary measures to ensure that the request parameter data is appropriately protected within the CS itself.

2.3.3 Configuration Request URI Example

Using the rules in Section 2.2, “[Obtaining the Configuration Service Domain](#)”, the UA has determined that the Configuration Service Domain value is "example.net". To obtain the base URL, the UA constructs the DNS NAPTR request for "example.net.", which returns the DNS records:

Example 2.1 Configuration Service NAPTR Query Results

```
NAPTR 10 10 "u" "SFUA.CFG" "!^.*$!https://p1.example.net/cfg!" ""
NAPTR 100 10 "u" "SFUA.CFG" "!^.*$!https://p2.example.net/cfg!" ""
NAPTR 90 50 "s" "SIP+D2T" "" _sip._tcp.example.net.
NAPTR 100 50 "s" "SIP+D2U" "" _sip._udp.example.net.
```

The records with the service-field "SFUA.CFG" each provide a base URL value for SIP UA configuration requests.

Our hypothetical example communications device is a 'HypoComm' version 2.1, made by ExampleCorp, and has the link layer MAC address of 00:11:22:33:44:55. It does not have any prior knowledge of a user identity for which to request configuration, so it constructs query parameters using the values it does have, combining each with the base URL to create these request URLs (lines wrapped for readability):

Example 2.2 Configuration Request URLs

```
https://p1.example.net/cfg
?sfua-id=urn:uuid:00000000-0000-1000-8000-001122334455
&sfua-vendor=examplecorp.com
&sfua-model=HypoComm
&sfua-revision=2.1
https://p2.example.net/cfg
?sfua-id=urn:uuid:00000000-0000-1000-8000-001122334455
&sfua-vendor=examplecorp.com
&sfua-model=HypoComm
&sfua-revision=2.1
```

2.4 Obtaining Configuration from the Configuration Service

To request configuration using a URL constructed as specified in Section 2.3, “[Constructing the Configuration Request URL](#)” the User Agent MUST do an HTTPS GET request to each of the URLs until a configuration that the UA can use is returned in response to one of the requests.

A successful final response from the Configuration Service to a GET request for configuration data MUST contain configuration data for the UA in the HTTP response body. Note that the full capabilities of HTTP are available to the CS, so responses such as redirection can be used by the CS as a part of the process of providing configuration data.

Configuration data returned in a successful response is subject to change by the CS. The HTTP cache control metadata (Etag or Last-Modified) returned in the response that provides configuration data is used to determine when a configuration change has occurred (Section 2.5.2, “[Configuration Change Notices](#)”) and to validate any stored configuration data (Section 2.6, “[Validity of Stored Configuration Data](#)”).

- An HTTP response from the CS that provides configuration data MUST include cache control metadata sufficient to ensure that when a new configuration is available, the cache control information for that new data is different.
- The UA MUST retain all of the HTTP cache control metadata from any response that provides configuration data.

2.4.1 Configuration Data Request Authentication

Because the Configuration Request URL scheme is HTTPS, the UA MUST always use TLS to establish a connection with the Configuration Service.

The UA MUST provide a `server_name` extension in the TLS Client Hello message as defined in [RFC3546] *Transport Layer Security (TLS) Extensions*, whose value is the host part of the CS base URL. This allows the CS to identify and provide a server certificate containing the desired identity (allowing for a single server to serve multiple names).

If the UA is capable of doing so, it SHOULD validate the server certificate provided by the CS. If the UA is unable to authenticate the certificate provided by the CS, it MAY store the server certificate and alert the user if that CS host provides a different certificate in the future. While not as secure as an authenticated certificate, this 'trust on first use' model does give some protection against 'man in the middle' attacks in the future.

If it has one, the UA MUST provide a client certificate. The CS SHOULD validate the UA client's certificate, if one is provided. If the CS is unable to authenticate the certificate provided by the UA (for example, the UA is using a self-signed certificate), then the CS MAY choose to cache the certificate, provided that the UA successfully authenticates using HTTP authentication (see next paragraph). This allows a CS to treat the digest authentication credentials as a single-use password to authenticate the client certificate. This 'trust on first use' model provides protection against future 'man in the middle' attacks, provided that the initial communication is not compromised.

If the CS requires HTTP authentication of the configuration data request, the HTTP 'username' parameter used MUST be the same value as the `sfua-user` value provided in the configuration data request parameters.

2.4.2 Configuration Data Request Failure

The HTTP configuration data request can fail in a number of ways; the error handling for each is defined below:

- If a DNS request to resolve the host name in the request URL returns a response that indicates that no matching result is available (NXDOMAIN), the UA MUST remove that request URL from the list of alternatives for the Configuration Service Domain.
- If the attempt to open a TCP connection to the host in the request URL fails, the UA MAY attempt requests to any alternative URLs for the same configuration service without waiting between alternatives, but any requests to the same host MUST wait between requests according to the procedure defined in Section 2.7, "Retry Backoff Procedure".
- If the TCP connection succeeds but the TLS handshake fails, including failure of the UA to validate the certificate provided by the Configuration Service host (if the UA is capable of validation), the UA MUST remove the failed URL from the list of alternative URLs for this Configuration Service Domain.
- If the request returns a permanent HTTP failure response, the UA MUST remove the failed URL from the list of alternatives for this Configuration Service Domain.
- If the list of alternatives for this Configuration Service Domain becomes empty, the UA MUST attempt to obtain another Configuration Domain Name using the procedures in Section 2.2, "Obtaining the Configuration Service Domain".
- If the UA has reached its chosen maximum number of retries (this Recommendation does not specify a maximum number of retries, but any retries to the same host MUST follow the procedure defined in Section 2.7, "Retry Backoff Procedure"), the UA MAY attempt to obtain another Configuration Domain Name using the procedures in Section 2.2, "Obtaining the Configuration Service Domain".

2.5 Requesting Configuration Change Notification

The configuration data provided by the CS is subject to change. In order to provide for the timely update of configuration information in User Agents, this Recommendation provides for notification of changes and refreshing of the configuration data.

2.5.1 Configuration Change Subscriptions

Any HTTP response from the CS that provides configuration data to the UA MUST include a Link header as specified by [draft-roach-sip-http-subscribe]; the URI value in the Link header MUST be a SIP URI, and the link relation ('rel' attribute) value MUST be 'monitor'. The 'monitor-group' relation MUST NOT be used - see below for rules regarding monitoring of multiple configuration data resources. The SIP URI returned in the Link header is the 'configuration change subscription URI'.

A UA that receives a successful configuration data response MUST attempt to maintain a subscription to the SIP URI from the Link header in that response for the http-monitor event package. This subscription is referred to herein as a 'configuration change subscription'.

The CS MUST accept properly authenticated SUBSCRIBE requests from the UA for the http-monitor event package at the URI it provided in the Link header of a configuration data response. Authentication of the SUBSCRIBE request uses any standard SIP authentication mechanism with credentials supplied to the UA in the configuration data.

Configuration data MAY include references in the form of additional URLs at the CS that the UA MUST use to obtain additional data. Any response to requests for these additional URLs that provide configuration data MUST provide cache control data and a configuration change subscription URI. The CS MAY return a unique configuration change subscription URI for each configuration data request, or MAY return the same SIP URI for different requests, so long as a change to the configuration data returned in any of these request results in notification on all subscriptions to the associated subscription URI.

If the CS returns a unique configuration change subscription URI in the Link header of different configuration data requests:

- The UA MUST maintain multiple subscriptions; one to each URI associated with configuration data the UA is using.

If the CS returns the same configuration change subscription URI in the Link header of different configuration data requests:

- The UA is NOT REQUIRED to create multiple subscriptions to the same URI.
- The UA MUST associate the URI with each of the configuration data requests for which it was returned, and any NOTIFY or other change in the status of that subscription affects the validity of all of the associated configuration data.
- The CS MUST send a NOTIFY message on the configuration change subscription when there is a change to any of the different configuration data resources for which the subscription URI was returned.

2.5.1.1 Change Subscription Failure

If a configuration change SUBSCRIBE request (either the initial request or any attempt to refresh the subscription) is permanently rejected by the Configuration Service (the CS returns a failure response that is not an authentication challenge or redirection and does not specify a Retry-After header), the UA MUST consider the associated configuration data to be not valid and attempt to revalidate it as specified in Section 2.6.1, "Re-validating Configuration Data". *Since the CS is not allowed to reject a properly authenticated request, this indicates a problem either with the configuration data or the CS.*

If a configuration change SUBSCRIBE request (either the initial request or any attempt to refresh the subscription) fails other than by being permanently rejected, the UA MUST consider the associated configuration data to be of unknown validity, and MUST retry the SUBSCRIBE request as specified in Section 2.7, "Retry Backoff Procedure"; the maximum time between retries MUST NOT be more than 30 minutes, and the retries MUST continue as long as the configuration is used. The UA MAY at any time return to any earlier step in the process of obtaining configuration data.

2.5.1.2 Change Subscription Termination

If the CS explicitly terminates the configuration change (http-monitor) subscription by sending a NOTIFY message with a Subscription-State header value of 'terminated', the UA MUST consider the configuration data to be of unknown validity. If the rules for interpreting and acting on the 'reason' code parameter as specified in section 3.2.4 of RFC 3265 allow, the UA MUST attempt to re-establish the subscription. If those rules do not allow the UA to re-subscribe, then the UA MUST consider the data to be not valid and attempt to revalidate it as specified in Section 2.6.1, "Re-validating Configuration Data". The UA MAY at any time return to any earlier step in the process of obtaining configuration data.

2.5.2 Configuration Change Notices

To inform the UA of a configuration data change, the CS MUST send a NOTIFY message to the UA in the configuration change subscription established by the UA as detailed in Section 2.5.2, "Configuration Change Notices".

The CS MUST NOT send unsolicited (out-of-dialog) NOTIFY messages.

As specified in [draft-roach-sip-http-subscribe], the body of a NOTIFY message in the http-monitor event package is the HTTP headers that would have been returned in response to an HTTP HEAD request (a HEAD request returns the headers that would have been returned for a GET request to the same URI, but with no body).

When a NOTIFY message is received by the UA in the configuration change subscription, the UA MUST compare the cache control data it retained when the configuration data was received with the HTTP header values in the NOTIFY message body. If any of the cache control data in the HTTP header values differs from those in the original configuration data response, the UA MUST consider the stored configuration data to be no longer valid. As soon as reasonably possible after the UA discovers that configuration data is no longer valid, the UA MUST attempt a GET request to the HTTPS configuration request URL which provided the configuration data to obtain the changed configuration data.

If this HTTPS request to the URL that previously provided the configuration data fails, the UA MUST attempt to obtain a new URL as specified in Section 2.3, “Constructing the Configuration Request URL”.

2.6 Validity of Stored Configuration Data

The configuration data stored by a UA is considered valid so long as the UA has a subscription to the CS as described in Section 2.5.1, “Configuration Change Subscriptions” and the UA has not received a NOTIFY message from the CS indicating that the configuration data has changed.

When a UA initializes itself, it MUST consider any stored configuration data to be of unknown validity.

The UA MAY use configuration data that is of unknown validity, or configuration data that is known to be no longer valid, while attempting to revalidate that data or obtain new data. There is no assurance that such configuration data is still useful, but the UA is NOT REQUIRED to stop using or delete the data.

2.6.1 Re-validating Configuration Data

To revalidate stored configuration data of unknown validity, the UA MUST repeat the HTTP GET request it used to obtain the stored configuration data, with the appropriate HTTP headers to make the request a conditional request using the cache control data returned in the response that provided the configuration data. This allows the CS to respond either with a new configuration data response, or a 304 (Not Modified) response to indicate that the configuration data has not changed.

If the CS responds with a 304 response, the SIP UA MUST assume that the value of the Link header returned with the original configuration data is also still correct (in effect, the HTTP cache control values and the subscription URL are a part of the configuration data), and so the UA MUST attempt to create and maintain a subscription to that URL as when the configuration data was first obtained (Section 2.5.1, “Configuration Change Subscriptions”).

If the HTTP request to revalidate the configuration fails, the UA MUST follow the procedures defined for a failure of the initial HTTP configuration data request as specified in Section 2.4.2, “Configuration Data Request Failure”.

2.7 Retry Backoff Procedure

In case of certain possible failures as described above, the appropriate response is to retry the failed operation. In all of these retry cases, the following rules apply:

- The UA SHOULD retry at least 5 times before abandoning the failed step (except as allowed for in specific error handling rules above).
- Following the first instance of a given failure, the UA MUST select an initial backoff timer value randomly between 2 and 8 inclusive and wait this number of seconds before retrying the failed request.
- Following any subsequent instance of a given failure, the UA MUST increase the backoff timer value by 2 raised to the power of the number of preceding failures (2^N where N is the number of previous failures), and wait this increased number of seconds or the maximum interval specified by specific error handling procedures, whichever is less, before retrying the failed request.

For example, after an initial failure, the UA chooses an initial backoff timer value of 4 seconds, followed by retries at the following times: 6 seconds ($4 + 2^1$), 10 seconds ($6 + 2^2$), 18 seconds ($10 + 2^3$), 32 ($18 + 2^4$) seconds, and 64 ($32 + 2^5$) seconds.

3 Configuration Data

This Recommendation does not specify the form or content of configuration data. As such, the contents of this section are non-normative.

3.1 Configuration Data Items

The configuration data for a SIP UA should, at minimum, include items with the following semantics.

3.1.1 Address-of-Record

The Address-of-Record (AOR) is a SIP or SIPS URI which identifies the user of the device as specified in [RFC 3261](#).

3.1.2 Realm

The realm is used to populate the realm parameter in the SIP Proxy-Authorization header as specified in [RFC 3261](#) when the UA receives an authentication challenge.

3.1.3 Username

The username is used to populate the username parameter in the SIP Proxy-Authorization header as specified in [RFC 3261](#) when the UA receives an authentication challenge.

3.1.4 Digest

The digest is a string containing the digest of the username, realm and password as specified in [RFC 2617](#) and is used to generate a response to an authentication challenge as specified in [RFC 3261](#).

3.1.5 OutboundProxy

The OutboundProxy if defined contains the default outbound proxy through which SIP requests, not explicitly routed, are routed as specified in [RFC 3261](#).

3.2 Reset User Agent to Default Configuration

The earlier sections of this Recommendation define methods by which the UA can be automatically provisioned. Some User Agents allow certain user specific settings (e.g. Contact Directory, specialized ring-tones etc.) to be set by a user, and possibly stored locally in the User Agent. Since it may be necessary to later re-assign a UA, designers of configuration data formats may want to provide for explicit controls for any such locally configured settings, including the ability to explicitly delete them to return the UA to a completely unconfigured state.

4 Internet Draft References

[draft-nottingham-http-link-header] Mark Nottingham, *Web Linking*, [Internet Engineering Task Force](#) .

[draft-roach-sip-http-subscribe] Adam Roach, Tekelec, *A SIP Event Package for Subscribing to Changes to an HTTP Resource*, [Internet Engineering Task Force](#) .

[draft-sipforum-ua-config] Michael Procter, VoIP.co.uk Scott D. Lawrence, Avaya Inc., *IANA Registration of the SIP Forum User Agent Configuration Application Service Tag*, [Internet Engineering Task Force](#) .

5 Referenced Standards

- [ANSI.TIA-1057-2006] *Telecommunications IP Telephony Infrastructure Link Layer Discovery Protocol for Media Endpoint Devices*, American National Standards Institute, April 1993.
- [RFC1034] P Mockapetris, Information Sciences Institute (ISI), *Domain Names - Concepts and Facilities*, Internet Engineering Task Force, November 1 1987.
- [RFC2119] Scott Bradner, Harvard University, *Key words for use in RFCs to Indicate Requirement Levels*, Internet Engineering Task Force, March 1997.
- [RFC2131] Ralph Droms, Bucknell University, Computer Science Department, *Dynamic Host Configuration Protocol*, Internet Engineering Task Force, March 1997.
- [RFC2132] Steve Alexander, Silicon Graphics, IncRalph Droms, Bucknell University, Computer Science Department, *DHCP Options and BOOTP Vendor Extensions*, Internet Engineering Task Force, March 1997.
- [RFC2396] T. Berners-Lee, World Wide Web ConsortiumR.T. Fielding, Department of Information and Computer Science, U.C. IrvineL. Masinter, Xerox PARC, *Uniform Resource Identifiers (URI): Generic Syntax*, Internet Engineering Task Force, March 1997.
- [RFC2616] Roy Fielding, >Department of Information and Computer ScienceJames Gettys, World Wide Web Consortium-Jeffrey C. Mogul, Compaq Computer CorporationHenrik Frystyk Nielsen, World Wide Web ConsortiumLarry Masinter, Xerox CorporationPaul J. Leach, Microsoft CorporationTim Berners-Lee, World Wide Web Consortium, *Hypertext Transfer Protocol -- HTTP/1.1*, Internet Engineering Task Force, June 1999.
- [RFC2617] John Franks, Northwestern University, Department of MathematicsPhillip M. Hallam-Baker, Verisign Inc.Jeffery L. Hostetler, AbiSource, Inc.Scott D. Lawrence, Agranat Systems, Inc.Paul J. Leach, Microsoft CorporationAri Luotonen, Netscape Communications CorporationLawrence C. Stewart, Open Market, Inc., *HTTP Authentication: Basic and Digest Access Authentication*, Internet Engineering Task Force, June 1999.
- [RFC3261] J. RosenbergH. SchulzrinneG. CamarilloA. JohnstonJ. PetersonR. SparksM. HandleyE. Schooler, *SIP: Session Initiation Protocol*, Internet Engineering Task Force, June 2002.
- [RFC3265] A.B. Roach, *Session Initiation Protocol (SIP)-Specific Event Notification*, Internet Engineering Task Force, June 2002.
- [RFC3315] R. DromsJ. BoundB. VolzT. LemonC. PerkinsM. Carney, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, Internet Engineering Task Force, July 2003.
- [RFC3319] H. SchulzrinneB. Volz, *Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers*, Internet Engineering Task Force, July 2003.
- [RFC3546] S. Blake-WilsonM. NystromD. HopwoodJ. MikkelsenT. Wright, *Transport Layer Security (TLS) Extensions*, Internet Engineering Task Force, June 2003.
- [RFC3958] L. DaigleA. Newton, *Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)*, Internet Engineering Task Force, January 2005.
- [RFC3986] Tim Berners-Lee, World Wide Web ConsortiumRoy Fielding, Day SoftwareLarry Masinter, Adobe Systems Incorporated, *Uniform Resource Identifier (URI): Generic Syntax*, Internet Engineering Task Force, January 2005.
- [RFC4848] L. Daigle, *Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)*, Internet Engineering Task Force, April 2007.
- [RFC5626] Cullen JenningsRohan MahyF. Audet, *Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)*, Internet Engineering Task Force, October 2009.

A Contributing Members of the SIP Forum User Agent Configuration Working Group

Francois Audet, Nortel Networks Inc.
Eric Burger, SIP Forum
Sumanth Channabasappa, Cable Television Laboratories, Inc. (CableLabs)
Martin Dolly, AT&T Labs
John Elwell, Siemens Enterprise Communications
Marek Dutkiewicz, Polycom Inc.
Andy Hutton, Siemens Enterprise Communications
Lincoln Lavoie, University of New Hampshire
Scott Lawrence, Avaya Inc.
Paul Mossman, Avaya Inc.
Michael Procter, VoIP.co.uk
Marc Robins, SIP Forum
Henning Schulzrinne, Columbia University
Rifaat Shekh-Yusef, Avaya Inc.
Robert Sparks, Tekelec
Simo Veikkolainen, Nokia

The Editor would like to also acknowledge valuable contributions by Leslie Daigle and Margaret Wasserman.

B Numeric Identifiers for SIP Service Providers

This appendix is non-normative.

The mechanism described in this appendix is designed to provide a means by which the Configuration Service Domain may be specified as a numeric string (so that only a telephone keypad is needed) by a user. Deployment of this mechanism requires the creation of a public highly available registry with all the administrative and legal complexity that entails. Since the establishment of such a registry will require additional time, and the remainder of this Recommendation is otherwise ready for implementation, the SIP Forum has decided to issue version 1.0 of this Recommendation without this mechanism, while continuing the process of defining and establishing the required registry for providers.

The SIP Forum expects that when a registry is established, an updated revision of this Recommendation will be issued that moves the material in this appendix back into the normative specification. It is of course possible that some details, such as the fixed DNS root used to construct the NAPTR query, may be modified at that time.

Provider Alias Number (PAN)

A string consisting of decimal digits used to identify a domain from which configuration is to be requested.

B.1 Numeric Domain Alias Entry

A UA MAY have an interface by which a string consisting only of decimal digits, called a *Provider Alias Number* (PAN), is provided by the user. The PAN value is resolved to the Configuration Service Domain using the S-NAPTR (Simple NAPTR) DDDS application [RFC3958].

The lookup key for the S-NAPTR request is the Provider Alias Number concatenated with the fixed suffix ".pan.sipforum.org." to form a DNS domain name. The UA MUST make a DNS request for NAPTR records for that domain name. From the returned records (see next paragraph), the UA MUST select the record whose Service field value is "SFUA.PAN"; the Configuration Service Domain Name is the value found in the Replacement field of this record.

The DNS for pan.sipforum.org. MUST be configured such that at most one NAPTR record is returned for any given Provider Alias Number, and MUST be configured to return the Flag field set to "a", an empty Regular Expression field, and the Substitution value set to the Configuration Service Domain Name.

B.1.1 Provider Alias Number Resolution Example

Through some manual process, the UA has obtained the provider alias number "555". To obtain the Configuration Service Domain Name, the UA constructs a DNS NAPTR request by appending the domain suffix ".pan.sipforum.org." to form the query key "555.pan.sipforum.org.", which returns the DNS record:

Example B.1 PAN NAPTR Query Result

```
NAPTR 10 10 "a" "SFUA.PAN" "" "example.net."
```

The record with the service-field "SFUA.PAN" provides the DNS name "example.net." for the Configuration Service Domain Name.

DRAFT