

SDP¹ parser requirements for Solaris SIP stack

Girish Moodalbail

Girish.Moodalbail@sun.com

Aug 5th 2007

Contents

1	Introduction	2
2	Competitive Analysis.....	3
3	Proposal	3
3.1	SDP Parsing:	3
3.2	SDP Syntax checking:	3
3.3	SDP generation:	3
4	References	3

¹ Session Description Protocol (RFC 4566)

1 Introduction

Session Description Protocol (SDP)ⁱ, as the name suggests, is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP conveys media details like type (audio, video, etc.) and encoding (MPEG video, DivX video, etc), transport protocol (RTP/UDP/IP, H.320, etc.), addresses and ports, session name, purpose and owner and other session description metadata to the participants. However it is not intended to support negotiation of session content or media encodings.

The Session Initiation Protocol (SIP)ⁱⁱ is an application-layer control protocol for creating, modifying, and terminating sessions such as Internet multimedia conferences, Internet telephone calls, and multimedia distribution. The SIP messages used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. These session descriptions are commonly formatted using SDP.

An SDP session description uses text encoding and consists of a number of lines of text of the form:

<type>=<value>

where *<type>* is a single lower-case letter and *<value>* is structured text whose format depends on *<type>*. In general, *<value>* is either a number of fields delimited by a single space character or a free format string, and is case-significant unless a specific field defines otherwise. To simplify parsing the lines of text need to be in a particular order. An example SDP description, from RFC 4566, is as shown below

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

The current Solaris SIP stack does not provide any support for parsing, generating and syntax checking of the SDP description inside SIP messages. The applications have to create a buffer filled with SDP description and call `sip_add_content()` to add SDP description to a SIP message and to retrieve the SDP description from SIP messages, applications need to use `sip_get_content()`. They have to parse the description and check for syntax themselves.

The aim of this document is to list requirements for generating, parsing and syntax checking of the SDP and then integrate them into the Solaris SIP stack.

2 Competitive Analysis

Several open source SIP stacks PJSIPⁱⁱⁱ, Sofia^{iv} and oSIP^v already provide support for parsing, generating and syntax checking of the SDP description. We need to add this feature in to our SIP stack to be on par with others and enhance the feature set of our stack.

3 Proposal

Solaris SIP stack currently does not provide support for parsing, generating and syntax checking of the SDP message within SIP request. Various Open Source SIP stack already provide support for parsing, generating and syntax checking as explained in the competitive analysis. We need to enhance our SIP stack with SDP parsing to increase adoption of our stack among the community. Solaris SIP stack must meet the following list of requirements.

3.1 SDP Parsing:

3.1.1 The Solaris SIP stack needs to parse the SDP message and provide access into each of the constituent field values through API's.

3.2 SDP Syntax checking:

3.2.1 The Solaris SIP stack should check for correctness of syntax for all the fields in the SDP message. Syntax checking should be compliant with Section 9 (SDP Grammar) of RFC 4566, so that we are interoperable with other SIP stacks.

3.2.2 The order of fields within the SDP description should also be checked.

3.3 SDP generation:

3.3.1 Provide API's to generate the constituent fields of the SDP messages.

4 References

ⁱ Session Description Protocol – RFC 4566

ⁱⁱ Session Initiation Protocol – RFC 3261

ⁱⁱⁱ www.pjsip.org

^{iv} <http://sofia-sip.sourceforge.net/refdocs/sdp/index.html>

^v <http://savannah.gnu.org/projects/osip/>