



## CHAPTER 4

# SDP API

このセクションでは、セッション記述プロトコル (SDP) のコンテンツ本文に関連付けられた API について説明します。SIP メッセージ オブジェクトとは異なり、SDP は基本的な文字列として扱われます。これを使用して、スクリプト作成者は Lua 文字列ライブラリの機能を利用することができます。Lua 文字列ライブラリ関数に加えて、シスコは、SDP の操作を容易にするためにいくつかの API も提供します。Cisco API は文字列ライブラリに直接追加されます。

ただし、スクリプトで SDP を操作するには、まず SIP メッセージ オブジェクトで提供される `getSdp()` API を使用して、Lua SIP メッセージ オブジェクトから SDP コンテンツ本文を取得する必要があります。その後は、スクリプトでシスコの API を含む文字列ライブラリを使用して、SDP を操作できます。変更すると、SIP メッセージ オブジェクトによって提供される `setSdp(sdp)` API を使用して、SIP メッセージ オブジェクトに SDP が書き戻されます。これらの API に関する詳細については、[SIP メッセージ API](#) を参照してください。

```
-- Get the SDP content body from the SIP message
local sdp = msg:getSdp()
-- modification of the SDP happens at this point
-- Update the SDP associated with the SIP message
msg:etSdp(sdp)
```

SDP を操作するために、シスコでは、個々の回線の取得、変更、追加、挿入、削除のための API、ならびにすべてのメディア記述を提供します。この章の以降の部分では、Cisco API について説明します。

SDP の任意の行を取得して操作するには、行を識別する手段が必要です。スクリプト作成者は、無効な SDP を作用するように対処することが求められます。したがって、正規化のために SDP の構造を知っておくことが最低限必要です。不可能ではないにしても、無効な SDP を作用させるのは簡単ではありません。有効な SDP を操作することも可能です。これらの目標を念頭に置いて、文字列ベースのアプローチが保証されています。

SDP を操作するための Lua API は、大きく分けると 2 つのカテゴリに分類されます。

- [行レベルの API](#) : この一連の API は、SDP 内の行を取得して操作できるようになっています
- [メディア記述レベルの API](#) : この一連の API は、特定のメディア記述を操作するためのものです。

セッション レベルの行の操作には、行レベルの一連の API を使用し、メディア記述の操作には、メディア記述レベルの一連の API を使用します。



(注) 行を操作するために設計された一連の API の操作対象は、セッション レベルの行のみに限定されるわけではありません。

## 行レベルの API

### Cisco API

次の行レベルの Cisco API が使用可能です。

- [getLine](#)( start-of-line , line-contains)
- [modifyLine](#)(start-of-line, line-contains ,new- line )
- [addline](#)(new-line)
- [insertLineAfter](#)( start-of-line, line-contains, new-line)
- [insertLineBefore](#)( start-of-line, line-contains, new-line)
- [removeLine](#)(start-of-line, line-contains)

#### パラメータの説明

パラメータ	説明
sdp	<b>sdp</b> パラメータは、SDP の入った文字列です。制御文字 (「\r」、「\n」 など) もすべて含みます。API では、sdp パラメータの SDP 構成を強制することはなく、文字列同様に扱います。スクリプト作成者は、API を使用して、SDP の一部分 (たとえば、音声メディア記述など) を操作できます。
start-of-line	<b>start-of-line</b> パラメータは、sdp 内の行の先頭と一致させるために使用する文字列です。一般的なスクリプトでは、たとえば、「o=」「a=rtpmap:9」を使用して、この文字列で始まる SDP の最初の行を取得/変更/挿入/削除できます。この文字列で始まる行が複数存在する場合は、最初の行だけが操作対象となります。  このパラメータは、<char>=<string> という形式になっていなければなりません。つまり、2 文字目が必ず等号でなくてはなりません。このパラメータがこの形式になっていない場合は、「nil」が返されます。
line-contains	<b>line-contains</b> は、「start-of-line」パラメータで始まる行内で検索する、追加の一致条件として使用される文字列です。nil が指定されている場合、 <b>line-contains</b> は一致条件としては使用されません。その場合、 <b>start-of-line</b> パラメータだけが検索に使用されます。  <b>line-contains</b> が nil でない場合、 <b>start-of-line</b> と <b>line-contains</b> の両方が検索に使用されます。(両方) 検索条件に一致する SDP の行が複数ある場合、最初の一致だけが返されます。  このパラメータで指定するこの追加の一致条件が必要ない場合は、値として「nil」を引用符なしで渡す必要があります。
new-line	<b>new-line</b> は、最終的な sdp (返された sdp) に追加/挿入する文字列です

上記の API ではパラメータを文字列として扱います。このため、セッション レベルの行とメディア レベルの行の両方を操作できます。行がセッション レベルかメディア レベルかに応じて動作が変わることは一切ありません。条件に一致する行を検索する際、各行は \n (または \r\n) で終わるものとします。

パラメータ内の文字はいずれも Lua の「マジック キャラクタ」とは見なされません。すべての文字が、特別な意味は一切持たず、プレーン文字として扱われます。

## getLine

`sdp:getLine(start-of-line, line-contains)` returns string

このメソッドでは、`sdp` 内の **start-of-line** で開始する行のうち、文字列 **line-contains** を含む最初の行が返されます。

**例：ローカル変数を、SDP からの c 行の値に設定します。**

### スクリプト

```
M = {}
function M.outbound_INVITE(msg)
  local sdp = msg:getSdp()
  if sdp
  then
    local ipv4_c_line = sdp:getLine("c=", "IP4")
  end
end
return M
```

### メッセージ

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.88
s=SIP Call
c=IN IP4 172.18.197.88
t=0 0
m=audio 24580 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

### 出力/結果

Local variable `ipv4_c_line` is set to `"c=IN IP4 172.18.197.88\r\n"`

## modifyLine

`sdp:modifyLine(start-of-line, line-contains, new-line)`

このメソッドは、**start-of-line** で始まる「`sdp`」の最初の行を検出します。**line-contains** パラメータが `nil` でない場合は、その値も行内に存在する必要があります。一致する行（行終了文字を含む）は **new-line** パラメータで置き換えられます。得られた `sdp` が返されます。

一致する行が見つからない場合は、元の `sdp` が未変更のまま返されます。



(注)

SDP のルールに従うために、**new-line** には `\r\n`（または `\n`）で終了する文字列を呼び出し側で指定することを推奨します。ユーティリティによって行終了文字が強制されることはありません。

**例：**次のコードは、a= 行の G.722 コーデックをドットなしの G722 に変更します。

### スクリプト

```
M = {}

function M.inbound_INVITE(msg)
    local sdp = msg:getSdp()

    if sdp
    then
        local g722_line = sdp:getLine("a=", "G.722")

        if g722_line
        then
            --Replace G.722 with G722. The dot is special and must be
            --escaped using % when using gsub.
            g722_line = g722_line:gsub("G%.722", "G722")
            sdp = sdp:modifyLine("a=", "G.722", g722_line)
            msg:setSdp(sdp)
        end
    end
end

return M
```

### メッセージ

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 23588 RTP/AVP 9 0 8 18 101
a=rtpmap:9 G.722/8000
a=ptime:20
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:8 PCMA/8000
a=ptime:20
a=rtpmap:18 G729/8000
a=ptime:20
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

### 出力/結果

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 23588 RTP/AVP 9 0 8 18 101
a=rtpmap:9 G722/8000
a=ptime:20
```

```

a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:8 PCMA/8000
a=ptime:20
a=rtpmap:18 G729/8000
a=ptime:20
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

## addline

```
sdp:addLine(new-line)
```

このメソッドは、SDPの末尾に **new-line** を追加します。SDPのルールに従うために、**new-line** には `\r\n` (または `\n`) で終了する文字列を呼び出し側で指定することを推奨します。このAPIによって行終了文字が強制されることはありません。

**例：SDPの末尾に属性行を追加します。**

### スクリプト

```

M = {}

function M.outbound_INVITE(msg)
    local sdp = msg:getSdp()

    if sdp
    then
        sdp = sdp:addLine("a=some-attribute\r\n")

        msg:setSdp(sdp)
    end
end

return M

```

### メッセージ

```

INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.88
s=SIP Call
c=IN IP4 172.18.197.88
t=0 0
m=audio 24690 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

### 出力/結果

```

INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.88

```

```
s=SIP Call
c=IN IP4 172.18.197.88
t=0 0
m=audio 24690 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=some-attribute
```

## insertLineAfter

```
sdp:insertLineAfter(start-of-line, line-contains, new-line)
```

このメソッドでは、**line-contains** パラメータが指定されている場合、sdp 内の **start-of-line** で開始する行のうち、**line-contains** も含む最初の行を検出します。次に、見つかった行の後に **new-line** を挿入します。

**例：SDP 内の「a=」および「G729」を持つ行の後に 1 行挿入します。**

### スクリプト

```
M = {}

function M.outbound_INVITE(msg)
    local sdp = msg.getSdp()

    if sdp
    then
        sdp = sdp:insertLineAfter("a=", "G729", "a=ptime:30\r\n")
        msg.setSdp(sdp)
    end
end

return M
```

### メッセージ

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 21702 RTP/AVP 18 101
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

### 出力/結果

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
```

```

c=IN IP4 172.18.197.29
t=0 0
m=audio 21702 RTP/AVP 18 101
a=rtpmap:18 G729/8000^M
a=ptime:30
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

## insertLineBefore

```
sdp:insertLineBefore(start-of-line, line-contains, new-line)
```

このメソッドでは、**line-contains** パラメータが指定されている場合、sdp 内の **start-of-line** で開始する行のうち、**line-contains** も含む最初の行を検出します。次に、見つかった行の前に **new-line** を挿入します。

**例：SDP 内の「s=」行の前に 1 行挿入します。**

### スクリプト

```

M = {}

function M.inbound_ANY_INVITE(msg)
    local sdp = msg:getSdp()

    if sdp
    then
        sdp= sdp:insertLineBefore("s=", nil, "e=nobody@cisco.com\r\n")
        msg:setSdp(sdp)
    end
end

return M

```

### メッセージ

```

SIP/2.0 200 OK
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 17774 RTP/AVP 9 101
a=rtpmap:9 G722/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

### 出力/結果

```

SIP/2.0 200 OK
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
e=nobody@cisco.com

```

```
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 17774 RTP/AVP 9 101
a=rtpmap:9 G722/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

## removeLine

```
sdp:removeLine(start-of-line, line-contains)
```

このメソッドでは、「line-contains」パラメータが指定されている場合、「sdp」内の「start-of-line」で開始する行のうち、「line-contains」を含む最初の行を検出します。一致した行は sdp から削除されます。

**例：「a=rtpmap」と「G729」の両方を含む行を SDP から削除します。**

### スクリプト

```
M = {}

function M.inbound_INVITE(msg)
    local sdp = msg.getSdp()

    if sdp
    then
        sdp = sdp:removeLine("a=rtpmap:", "G729")
        msg.setSdp(sdp)
    end
end

return M
```

### メッセージ

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
.
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 25328 RTP/AVP 9 0 8 18 101
a=rtpmap:9 G722/8000
a=ptime:20
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:8 PCMA/8000
a=ptime:20
a=rtpmap:18 G729/8000
a=ptime:20
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

### 出力/結果

```
INVITE sip:901@rawlings.cisco.com:5080 SIP/2.0
```

```

Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 2000 1 IN IP4 172.18.197.22
s=SIP Call
c=IN IP4 172.18.197.29
t=0 0
m=audio 25328 RTP/AVP 9 0 8 18 101
a=rtpmap:9 G722/8000
a=ptime:20
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:8 PCMA/8000
a=ptime:20
a=ptime:20
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

## メディア記述レベルの API

SDP メディア レベルでの SDP の取得と操作に使用される API は次のとおりです。

### SDP の例

次の SDP の例では、セッション レベルの行は緑色で示し、メディア レベルの行を青色で示していません。

```

v=0\r\n
  o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
  s=SIP Call\r\n
  c=IN IP4 172.18.195.126\r\n
  t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
  a=rtpmap:9 G722/8000\r\n
  a=ptime:20\r\n
  a=rtpmap:18 G729/8000\r\n
  a=ptime:20\r\n
  a=fmtp:18 annexb=no\r\n
  a=rtpmap:101 telephone-event/8000\r\n
  a=fmtp:101 0-1\r\n
[2] m=video...

```

上の SDP の例では、メディア レベルのインデックスは青色になっています。メディア レベル [1] は、音声メディアの記述です。メディア レベル [2] は、ビデオメディアの記述です（全体は示していません）。

インデックスはすべて、1 から始まります。したがって、最初のメディア記述はインデックス 1 です。2 つ目はインデックス 2 となり、以降も同様です。

メディア記述レベルでは、次の API が使用できます。

- [getMediaDescription\(media-level\)](#)
- [getMediaDescription\(media-contains\)](#)
- [modifyMediaDescription\(media-level, media-description\)](#)
- [modifyMediaDescription\(media-contains, media-description\)](#)
- [addMediaDescription\(media-description\)](#)

- `insertMediaDescription(media-level, media-description)`
- `removeMediaDescription(media-level)`
- `removeMediaDescription(media-contains)`

### パラメータの説明

パラメータ	説明
<code>sdp</code>	<code>sdp</code> オブジェクトは、SDP の入った文字列です。制御文字 (「 <code>\r</code> 」、「 <code>\n</code> 」など) もすべて含みます。
<code>media-level</code>	<code>media-level</code> パラメータは、SDP 内のメディア記述を参照するために使用されるインデックスです。インデックスは、1 から始まります。SDP 内の最初のメディア記述は、メディア レベル 1 です。2 目はメディア レベル 2 で、以降も同様です。
<code>media-contains</code>	<code>media-contains</code> パラメータは、 <code>m</code> 行内で一致対象を (例: メディア タイプ) 見つけるために使用する文字列です。  たとえば、 <code>audio</code> 、 <code>video</code> 、 <code>message</code> などを使用して、完全に一致する <code>media-contains</code> 値を <code>m</code> 行に含む、最初に一致するメディア記述を取得、修正、または削除できます。
<code>media-description</code>	<code>media-description</code> パラメータは、完全なメディア記述を示す文字列です。複数の行からなり必要な埋め込み制御文字 (「 <code>\r</code> 」および「 <code>\n</code> 」) を含みます。  最後の行は「 <code>\r\n</code> 」で終了させることを強く推奨します。ただし、これらの API が行う内容は純粋なテキスト操作で、テキストの形式に関して強制されるルールは一切ありません。実際に、実装によっては、各行の末尾に「 <code>\r</code> 」が含まれない場合もあります。これを純粋なテキストとして扱うことにより、いずれのケースにも対処するために必要な柔軟性を提供し、さらに 2 つのスタイル間で簡単に変換することもできます (たとえば、「 <code>\r\n</code> 」を「 <code>\n</code> 」に変換したり、その逆に変換したりすることはそれほど問題ではありません)。  次の文字列は、完全なメディア記述の一例です。  " <code>m=audio 18884 RTP/AVP 9\r\na=rtpmap:9 G722/8000\r\n</code> "

以降で、各メディア関連 API の詳細について説明します。



(注)

インデックス (つまり `media-level`) は、すべて 1 から始まります

## getMediaDescription(media-level)

`getMediaDescription(media-level)` returns `media-description`

この API は、指定したメディア レベルの特定のメディア記述を取得するのに使用します。

**例：ローカル変数を、最初のメディア記述の値に設定します。**

### スクリプト

```
M = {}
function M.outbound_INVITE(msg)
  local sdp = msg:getSdp()
  if sdp
  then
```

```

        local m1 = sdp:getMediaDescription(1)
    end
end
return M

```

### メッセージ

```

v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
s=SIP Call\r\n
c=IN IP4 172.18.195.126\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video...

```

### 出力/結果

Local variable m1 is set to a string

```

m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n

```

## getMediaDescription(media-contains)

getMediaDescription(media-contains) returns media-description

この API は、m 行に media-contains と完全に一致する部分を持つ特定のメディア記述を取得するために使用します。



(注) 複数の m 行が一致する場合は、media-description と一致する最初のメディア記述だけが返されます。

**例：**ローカル変数を、m 行に「audio」というテキストを含む最初のメディア記述の値に設定します。

### スクリプト

```

M = {}
function M.outbound_INVITE(msg)
    local sdp = msg:getSdp()
    if sdp
    then
        local audio = sdp:getMediaDescription("audio")
    end
end
return M

```

## メッセージ

```
v=0\r\n
  o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
  s=SIP Call\r\n
  c=IN IP4 172.18.195.126\r\n
  t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
    a=rtpmap:9 G722/8000\r\n
    a=ptime:20\r\n
    a=rtpmap:18 G729/8000\r\n
    a=ptime:20\r\n
    a=fmtp:18 annexb=no\r\n
    a=rtpmap:101 telephone-event/8000\r\n
    a=fmtp:101 0-1\r\n
[2] m=video...
```

## 出力/結果

Local variable audio is set to a string

```
m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
```

## modifyMediaDescription(media-level, media-description)

```
modifyMediaDescription(media-level, media-description)
```

この API は、指定したメディア レベルの特定のメディア記述を修正するのに使用します。

**例: 「a=ptime:20\r\n」を含む行を音声メディア記述から削除します**

## スクリプト

```
M = {}
function M.outbound_INVITE(msg)
  local sdp = msg:getSdp()
  if sdp
  then
    local m1 = sdp:getMediaDescription(1)
    if m1
    then
      m1 = m1:gsub("a=ptime:20\r\n", "")
      sdp = sdp:modifyMediaDescription(1, m1)
      msg:setSdp(sdp)
    end
  end
end
return M
```

## 正規化前のメッセージ

```
v=0\r\n
  o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
  s=SIP Call\r\n
  c=IN IP4 172.18.195.126\r\n
  t=0 0\r\n
```

```
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
    a=rtpmap:9 G722/8000\r\n
    a=ptime:20\r\n
    a=rtpmap:18 G729/8000\r\n
    a=ptime:20\r\n
    a=fmtp:18 annexb=no\r\n
    a=rtpmap:101 telephone-event/8000\r\n
    a=fmtp:101 0-1\r\n
[2] m=video...
```

### 正規化後のメッセージ

After normalization, the SDP would contain:

```
v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
s=SIP Call\r\n
c=IN IP4 172.18.195.126\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
    a=rtpmap:9 G722/8000\r\n
    a=rtpmap:18 G729/8000\r\n
    a=fmtp:18 annexb=no\r\n
    a=rtpmap:101 telephone-event/8000\r\n
    a=fmtp:101 0-1\r\n
[2] m=video...
```

## modifyMediaDescription(media-contains, media-description)

modifyMediaDescription(media-contains, media-description)

この API は、**m** 行に **media-contains** と完全に一致する部分を持つ特定のメディア記述を修正するのに使用します。複数の **m** 行が一致する場合は、**media-description** と一致する最初のメディア記述だけが修正されることに注意してください。

**例 : 「a=ptime:20\r\n」を含む行を音声メディア記述から削除します**

### スクリプト

```
M = {}
function M.outbound_INVITE(msg)
    local sdp = msg:getSdp()
    if sdp
    then
        local audio = sdp:getMediaDescription("audio")
        if audio
        then
            audio = audio:gsub("a=ptime:20\r\n", "")
            sdp = sdp:modifyMediaDescription("audio", audio)
            msg:setSdp(sdp)
        end
    end
end
return M
```

### 正規化前のメッセージ

```
v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
s=SIP Call\r\n
```

```

c=IN IP4 172.18.195.126\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video...

```

### 正規化後のメッセージ

```

v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 172.18.195.96\r\n
s=SIP Call\r\n
c=IN IP4 172.18.195.126\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video...

```

## addMediaDescription(media-description)

addMediaDescription(media-description)

この API は、1 つのメディア記述を最後のメディア記述として追加するのに使用します。

### 例：

このスクリプトは、着信 INVITE からビデオ メディア記述を削除して、保存します。このスクリプトでは、ビデオ メディア記述が 2 つ目のメディア記述であることを前提としています。対応する発信応答については、このスクリプトは、格納されているビデオ メディア記述を取得し、ポートを 0 に設定して、そのメディア記述を発信 SDP に追加します。

### スクリプト

```

M = {}
function M.inbound_INVITE(msg)
  local sdp = msg:getSdp()
  if sdp
  then
    local video = sdp:getMediaDescription(2)
    if video
    then
      --remove the video media description
      sdp = sdp:removeMediaDescription(2)
      --store video media description
      local context = msg:getContext()
      context["video"] = video
      msg:setSdp(sdp)
    end
  end
end
end

```

```

function M.outbound_ANY_INVITE(msg)
    local sdp = msg:getSdp()
    --assume other side is expecting video before audio when there is
    --a video m-line in the SDP
    if sdp
    then
        local context = msg:getContext()
        local video = context["video"]
        if video
        then
            --set port to zero in SDP answer
            video = video:gsub("video %d* RTP", "video 0 RTP")
            sdp = sdp:addMediaDescription(video)
            msg:setSdp(sdp)
        end
    end
end
return M

```

### 正規化前のメッセージ

```

INVITE SDP . . .

v=0\r\n
o=- 2000 3 IN IP4 10.10.10.100\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.100\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video 32068 RTP/AVP 112 113 114\r\n
b=TIAS:4000000\r\n
a=rtpmap:112 H264/90000\r\n
a=fmtp:112
profile-level-id=4D8028;packetization-mode=1;max-mbps=243000;max-fs=8100\r\n
a=rtpmap:113 H264/90000\r\n
a=fmtp:113 profile-level-id=42400D;packetization-mode=1;max-mbps=11880;max-fs=396\r\n
a=rtpmap:114 H264/90000\r\n
a=fmtp:114 profile-level-id=42400D;packetization-mode=0;max-mbps=11880;max-fs=396\r\n
a=sendrecv\r\n
\r\n

200 Ok SDP . . .

v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 10.10.10.1\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.200\r\n
t=0 0\r\n
[1] m=audio 16007 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
\r\n

```

## 正規化後のメッセージ

```

INVITE SDP . . .

    v=0\r\n
    o=- 2000 3 IN IP4 10.10.10.100\r\n
    s=SIP Call\r\n
    c=IN IP4 10.10.10.100\r\n
    t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
    a=rtpmap:9 G722/8000\r\n
    a=ptime:20\r\n
    a=rtpmap:18 G729/8000\r\n
    a=ptime:20\r\n
    a=fmtp:18 annexb=no\r\n
    a=rtpmap:101 telephone-event/8000\r\n
    a=fmtp:101 0-1\r\n
    \r\n

200 Ok SDP . . .

    v=0\r\n
    o=CiscoSystemsCCM-SIP 2000 3 IN IP4 10.10.10.1\r\n
    s=SIP Call\r\n
    c=IN IP4 10.10.10.200\r\n
    t=0 0\r\n
[1] m=audio 16007 RTP/AVP 9 18 101\r\n
    a=rtpmap:9 G722/8000\r\n
    a=ptime:20\r\n
    a=rtpmap:101 telephone-event/8000\r\n
    a=fmtp:101 0-1\r\n
[2] m=video 0 RTP/AVP 112 113 114\r\n
    b=TIAS:4000000\r\n
    a=rtpmap:112 H264/90000\r\n
    a=fmtp:112
profile-level-id=4D8028;packetization-mode=1;max-mbps=243000;max-fs=8100\r\n
    a=rtpmap:113 H264/90000\r\n
    a=fmtp:113 profile-level-id=42400D;packetization-mode=1;max-mbps=11880;max-fs=396\r\n
    a=rtpmap:114 H264/90000\r\n
    a=fmtp:114 profile-level-id=42400D;packetization-mode=0;max-mbps=11880;max-fs=396\r\n
    a=sendrecv\r\n
    \r\n

```

## insertMediaDescription(media-level, media-description)

```
insertMediaDescription(media-level, media-description)
```

この API は、指定したメディア レベルのメディア記述を追加するのに使用します。

**例：メッセージ メディア記述を、それが SDP 内の最初のメディア記述になるように挿入します**

### スクリプト

```

M = {}
function M.outbound_INVITE(msg)
    local sdp = msg:getSdp()
    if sdp
    then
        local message = "m=message 1234 sip:alice@10.10.10.100\r\n"
        sdp = sdp:insertMediaDescription(1, message)
        msg:setSdp(sdp)
    end
end
end

```

```
return M
```

### 正規化前のメッセージ

```
INVITE SDP . . .

v=0\r\n
o=- 2000 3 IN IP4 10.10.10.100\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.100\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
```

### 正規化後のメッセージ

```
INVITE SDP . . .

v=0\r\n
o=- 2000 3 IN IP4 10.10.10.100\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.100\r\n
t=0 0\r\n
[1] m=message 1234 sip:alice@10.10.10.100\r\n
[2] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
```

## removeMediaDescription(media-level)

```
removeMediaDescription(media-level)
```

この API は、指定したメディア レベルのメディア記述を削除するのに使用します。

例：

[addMediaDescription\(media-description\)](#) で提供されている例を参照してください。この例は、`removeMediaDescription` も使用しています。

## removeMediaDescription(media-contains)

```
removeMediaDescription(media-contains)
```

この API は、`m` 行に `media-contains` と完全に一致する部分を持つ特定のメディア記述を削除するのに使用します。



(注)

複数の `m` 行が一致する場合は、`media-description` と一致する最初のメディア記述だけが削除されます。

**例：**

このスクリプトは、着信 INVITE からビデオ メディア記述を削除して、保存します。対応する発信応答については、このスクリプトは、格納されているビデオ メディア記述を取得し、ポートを 0 に設定して、そのメディア記述を発信 SDP に追加します。

**スクリプト**

```
M = {}
function M.inbound_INVITE(msg)
    local sdp = msg:getSdp()
    if sdp
    then
        local video = sdp:getMediaDescription("video")
        if video
        then
            --remove the video media description
            sdp = sdp:removeMediaDescription("video")
            --store video media description
            local context = msg:getContext()
            context["video"] = video
            msg:setSdp(sdp)
        end
    end
end

function M.outbound_ANY_INVITE(msg)
    local sdp = msg:getSdp()
    --assume other side is expecting video before audio when there is
    --a video m-line in the SDP
    if sdp
    then
        local context = msg:getContext()
        local video = context["video"]
        if video
        then
            --set port to zero in SDP answer
            video = video:gsub("video %d* RTP", "video 0 RTP")
            sdp = sdp:addMediaDescription(video)
            msg:setSdp(sdp)
        end
    end
end
return M
```

**正規化前のメッセージ**

```
INVITE SDP . . .

v=0\r\n
o=- 2000 3 IN IP4 10.10.10.100\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.100\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
a=ptime:20\r\n
a=rtpmap:18 G729/8000\r\n
a=ptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video 32068 RTP/AVP 112 113 114\r\n
b=TIAS:4000000\r\n
```

```

a=rtpmap:112 H264/90000\r\n
a=fmtp:112
profile-level-id=4D8028;packetization-mode=1;max-mbps=243000;max-fs=8100\r\n
a=rtpmap:113 H264/90000\r\n
a=fmtp:113 profile-level-id=42400D;packetization-mode=1;max-mbps=11880;max-fs=396\r\n
a=rtpmap:114 H264/90000\r\n
a=fmtp:114 profile-level-id=42400D;packetization-mode=0;max-mbps=11880;max-fs=396\r\n
a=sendrecv\r\n
\r\n

200 Ok SDP . . .

v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 10.10.10.1\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.200\r\n
t=0 0\r\n
[1] m=audio 16007 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
aptime:20\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
\r\n

```

### 正規化後のメッセージ

INVITE SDP . . .

```

v=0\r\n
o=- 2000 3 IN IP4 10.10.10.100\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.100\r\n
t=0 0\r\n
[1] m=audio 18884 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
aptime:20\r\n
a=rtpmap:18 G729/8000\r\n
aptime:20\r\n
a=fmtp:18 annexb=no\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
\r\n

200 Ok SDP . . .

v=0\r\n
o=CiscoSystemsCCM-SIP 2000 3 IN IP4 10.10.10.1\r\n
s=SIP Call\r\n
c=IN IP4 10.10.10.200\r\n
t=0 0\r\n
[1] m=audio 16007 RTP/AVP 9 18 101\r\n
a=rtpmap:9 G722/8000\r\n
aptime:20\r\n
a=rtpmap:101 telephone-event/8000\r\n
a=fmtp:101 0-1\r\n
[2] m=video 0 RTP/AVP 112 113 114\r\n
b=TIAS:4000000\r\n
a=rtpmap:112 H264/90000\r\n
a=fmtp:112
profile-level-id=4D8028;packetization-mode=1;max-mbps=243000;max-fs=8100\r\n
a=rtpmap:113 H264/90000\r\n
a=fmtp:113 profile-level-id=42400D;packetization-mode=1;max-mbps=11880;max-fs=396\r\n
a=rtpmap:114 H264/90000\r\n

```

```
a=fmtp:114 profile-level-id=42400D;packetization-mode=0;max-mps=11880;max-fs=396\r\na=sendrecv\r\n\r\n
```