



# CHAPTER 8

## Trace API

スクリプト トレーシングを使用して、スクリプト作成者がスクリプト内からトレースを生成できます。これは、スクリプトのデバッグだけに使用してください。スクリプトの動作を開始する場合は、これを無効にする必要があります。スクリプト トレーシングを有効にする方法は 2 通りあります。

1. 設定による方法：SIP トランクまたは SIP プロファイルに関連付けられた [スクリプト トレース (Script Trace)] チェックボックスを参照してください。
2. スクリプトを使用する：`trace.enable()` API を参照してください。

次の表に、上述の 2 つの設定のすべての組み合わせにおいて、トレースがスクリプトによって生成されるかどうかを示します。

設定	スクリプト	生成されるトレース
有効	有効	Yes
有効	無効	No
無効	有効	No
無効	無効	No

トレース ライブラリには、次の API があります。

- [trace.format](#)
- [trace.enable](#)
- [trace.disable](#)
- [trace.enabled](#)

### trace.format

```
trace.format(format-string, p1, p2, ...)
```

この関数は、フォーマット文字列とパラメータのリストが指定された場合、スクリプトに関連付けられているトランクのトレーシングが有効になっていて、そのスクリプトで `trace.enable()` によってトレーシングが有効にされていれば、パラメータをフォーマット文字列に挿入し、その出力を SDI トレース ファイルに書き出します。

#### 例

```
M = {}
trace.enable()
function M.inbound_INVITE(msg)
    local callid = msg.getHeader("Call-Id")
```

```

        trace.format("Call-Id header is %s", callid)
    end
    return M

```

## trace.enable

```
trace.enable()
```

この関数は、スクリプトでトレーシングをローカルに有効にできます。



**(注)** トレーシングは、スクリプトを使用して、SIP トランクまたは SIP プロファイルの設定によって有効になっている必要もあります。

### 例

前述の例、[trace.format](#) を参照してください。

## trace.disable

```
trace.disable()
```

この関数は、スクリプトでトレーシングをローカルに無効にできます。



**(注)** 関連付けられている SIP トランクまたは SIP プロファイルのスクリプト トレース設定フラグがオンになっている場合は、たとえここで無効にしても、Trace.output によってスクリプト トレースが生成されます。

### 例：

```

M = {}
trace.disable()
function M.inbound_INVITE(msg)
    local callid = msg.getHeader("Call-Id")
    trace.format("Call-Id header is %s", callid)
end
return M

```

## trace.enabled

```
trace.enabled() turns a boolean
```

スクリプトのこのインスタンスに関連付けられているデバイス上でスクリプト トレーシングが有効になっているか、またはスクリプト自身でトレーシングを有効にしていた場合に、**true** が返されます。トレーシングが有効になっていない場合は、**false** が返されます。

### 例

```

M = {}
trace.enable()
function M.inbound_INVITE(msg)
    if trace.enabled()
    then

```

```
        local callid = msg:getHeader("Call-Id")
        trace.format("Call-Id header is %s", callid)
    end
end
return M
```

