



CHAPTER 1

概要

このマニュアルでは、Cisco Unified CM- Session Management (SME) の SIP メッセージのカスタマイズに使用されるインターフェイス仕様について説明します。Cisco Unified CM-SME および API で使用可能な、SIP の透過性機能および正規化機能をサポートするための Lua 環境の詳細について説明します。

この章では、次のトピックについて説明します。

- [概要](#)
- [スクリプト環境](#)
- [メッセージハンドラ](#)

概要

この章では、特定の配置環境での、セッション管理 (SM) の動作のカスタマイズに使用されるインターフェイスである、Lua 環境について説明します。Lua は、非独占の、動作が軽いスクリプト言語です。このマニュアルの対象者は、Lua スクリプト言語の基本的な知識があることが前提となります。

Cisco Unified CM では、SIP メッセージをカスタマイズするための SIP 正規化および透過性と呼ばれる機能のセットが提供されます。

- 正規化：これは、着信および発信メッセージの変換プロセスです。
 - 着信メッセージでは、正規化は、ネットワークからメッセージを受信した後に発生します。着信メッセージの正規化は、メッセージを Cisco Unified CM により適合させるために使用されます。たとえば、Cisco Unified CM では、リダイレクト番号情報を伝達するための Diversion ヘッダーがサポートされます。Cisco Unified CM に接続される一部の SIP デバイスでは、この目的に History-Info ヘッダーが使用されます。着信の最適化では、Cisco Unified CM でリダイレクト情報が認識されるよう、History-Info ヘッダーが Diversion ヘッダーに変換されます。
 - 発信メッセージの場合、正規化は、ネットワークにメッセージを送信する直前に発生します。したがって、発信メッセージの正規化は、メッセージを外部 SIP デバイス (例：別の SIP 対応 PBX) により適合させるために使用されます。たとえば、発信の正規化は、外部 SIP デバイスでリダイレクト情報が認識されるよう、Diversion ヘッダーを History-Info ヘッダーに変換するために使用できます。
- 透過性：これを使用すると、Cisco Unified CM が Back to Back User Agent (B2BUA) の場合でも、1つのコールレグから別のコールレグに SIP 情報を渡すことができます。

正規化と透過性の機能は、SIP トランクまたは SIP 回線に関連付けられているスクリプトによって示されます。スクリプトは、発着信 SIP メッセージで動作するメッセージハンドラのセットとして示されます。正規化では、スクリプトによって、次のものを含む SIP メッセージのほとんどの状態が操作されます。

- 要求 URI
- 応答コードとフレーズ
- SIP ヘッダー
- SIP パラメータ
- コンテンツ本文
- SDP

透過性では、スクリプトによって、次のものを含む SIP メッセージのほとんどの情報が渡されません。

- SIP ヘッダー
- SIP パラメータ
- コンテンツ本文

このマニュアルでは、SIP メッセージ情報を操作し、渡すために使用される、スクリプト環境および API について説明します。

スクリプト管理の詳細については、『*Cisco Unified Communications Manager Administration Guide*』を参照してください。

スクリプト環境

Cisco Unified CM SIP トランクの動作をカスタマイズするためのインターフェイスは、Lua というスクリプト言語によって提供されます。Lua は、オープンソースの、動作が軽いスクリプト言語です。

Cisco Unified CM (SM) に使用できる Lua 環境は Lua の限定的なサブセットです。Lua で提供される基本機能の他に、次のような機能があります。

- 単語表記
- 値とタイプ
- 変数
- ステートメント
- 式など

Lua では、次のような一部の基本ライブラリも提供されています。

- ベース
- 共通ルーチン
- モジュール
- 文字列操作
- テーブル操作
- 数学関数
- OS 機能
- IO 機能

- デバッグ機能

ただし、SM で使用可能な Cisco SIP Lua 環境では、ベース ライブラリの全体およびサブセットの文字列ライブラリのみがサポートされます。他のライブラリはサポートされません。

ベース ライブラリでは、次がサポートされます。

- ipairs
- pairs
- next
- unpack
- error
- type
- tostring

Cisco SIP Lua 環境では、使用するスクリプト用にグローバル環境があります。この環境では、スクリプトに対してデフォルトの Lua グローバル環境（つまり `_G`）は提供されません。

Lua スクリプトでは、メッセージハンドラと呼ばれるコールバック機能のセットが提供され、SM 環境のコンテキストで SIP メッセージを操作します。メッセージハンドラの名前は、特定の SIP メッセージに対して起動されるハンドラを示します。たとえば、スクリプトの「`inbound_INVITE`」メッセージハンドラは、Cisco Unified CM で着信 INVITE を受信したときに起動されます。メッセージハンドラでは、SIP メッセージを表す `msg` と呼ばれる 1 つのパラメータを受信します。Lua スクリプトでは、シスコの SIP メッセージライブラリに定義されている API を使用して、`msg` パラメータにアクセスして操作を行います。

このマニュアルの以降では、メッセージハンドラの構造の詳細について説明します。次の項にシスコの SIP メッセージライブラリ API についての詳細を説明します。

ここでは、発信 INVITE で「Cisco-Guid」ヘッダーを削除するだけのスクリプトの例を示します。スクリプトの説明を簡単にするため、スクリプトの左側に行番号を示します。

簡単なスクリプト : M.lua

```

1. M = {}
2. function M.inbound_INVITE(msg)
3.     msg:convertHIToDiversion()
4. end
5. function M.outbound_INVITE(msg)
6.     msg:removeHeader("Cisco-Guid")
7. end
8. return M

```

上記のスクリプトに重要な部分が 3 つあります。

1. モジュールの初期化：スクリプトの最初の行で、「M」と呼ばれる Lua テーブルが作成され、空に初期化されます。このテーブルには、このスクリプトによって提供されるコールバック関数のセットが含まれます。変数 **M** は Lua テーブルで、モジュールの名前でもあります。
2. メッセージハンドラ定義：2～4 行目で、着信 INVITE のメッセージハンドラが定義されます。このスクリプトに関連付けられた SIP トランクまたは SIP 回線で着信 INVITE が受信されたときに、このコールバック関数が実行されます。この例では、メッセージハンドラによって API が起動され、History-Info ヘッダーが Diversion ヘッダーに変換されます。

5～6 行目で、発信 INVITE のメッセージハンドラが定義されます。このスクリプトに関連付けられた SIP トランクまたは SIP 回線で発信 INVITE が送信されたときに、このコールバック関数が実行されます。この例では、メッセージハンドラによって API が起動され、「Cisco-Guid」ヘッダーが削除されます。

スクリプトで複数のメッセージハンドラを定義できます。メッセージハンドラの名前は、特定の SIP メッセージに対して呼び出されるメッセージハンドラがあれば、そのハンドラを示します。

3. モジュールを返す：最後の行はモジュールの名前を返します。この行は必須です。これが、Cisco SIP Lua 環境で、スクリプトに関連付けられているメッセージハンドラを見つける方法です。

メッセージハンドラ

Lua スクリプトでは、メッセージハンドラと呼ばれるコールバック機能のセットが提供され、SM 環境のコンテキストで SIP メッセージを操作します。メッセージハンドラの名前は、特定の SIP メッセージに対して起動されるハンドラを示します。

名前付け

メッセージハンドラの命名規則によって、特定の SIP メッセージに対して起動されるメッセージハンドラが示されます。仕様による各種 SIP メッセージは、要求と応答に分かれます。

- 要求では、メッセージハンドラは、メッセージの方向と SIP 要求のメソッド名に従って命名されます。メソッド名は、SIP メッセージの「要求行」にある名前です。

```
<direction>_<method>
```

例

```
inbound_INVITE
outbound_UPDATE
```

- 応答では、メッセージハンドラは、メッセージの方向、応答コード、および SIP メソッドに従って命名されます。応答では、SIP メッセージの **CSeq** ヘッダーから、メソッド名が取得されます。

```
<direction>_<response code>_<method>
```

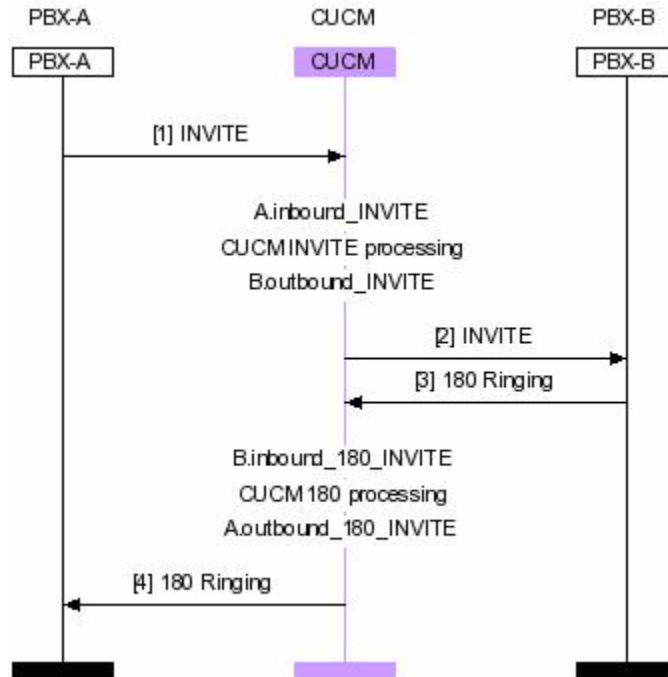
例

```
inbound_183_INVITE
inbound_200_INVITE
outbound_200_UPDATE
```

使用例

Cisco Unified CM-SME が 2 つのトランクを介して PBX-A および PBX-B に接続される場合について考えます。モジュール **A** を返すスクリプトは、PBX-A に接続されているトランクに接続されます。同様に、モジュール **B** を返すスクリプトは、PBX-B に接続されているトランクに接続されます。

次のハンドラが、INVITE ダイアログに対して実行されます。



ワイルドカード

メッセージハンドラ名は、ワイルドカードもサポートします。ワイルドカードのサポートは、メッセージが**要求 SIP**メッセージか**応答 SIP**メッセージかによって異なります。

要求メッセージ

ワイルドカード **ANY** は、<method> の場所に使用できます。<direction> では、ワイルドカードはサポートされません。

有効な要求メッセージハンドラ名については、[表 1-1 \(P.1-5\)](#) を参照してください。

表 1-1 有効な要求メッセージハンドラ名

メッセージハンドラ	説明
M.inbound_INVITE	このメッセージハンドラは、初期 INVITE および reINVITE を含むすべての着信 INVITE メッセージに対して起動されます。
M.inbound_ANY	このメッセージハンドラは、すべての着信要求に対して起動されます。
M.outbound_ANY	このメッセージハンドラは、すべての発信要求に対して起動されます。

応答メッセージ

ワイルドカード **ANY** は、<method> と <response code> のいずれか一方または両方の場所に使用できます。<direction> では、ワイルドカードはサポートされません。さらに、ワイルドカード文字 **X** は <response code> で使用できます。

有効な応答メッセージハンドラ名については、[表 1-2 \(P.1-6\)](#) を参照してください。

表 1-2 有効な応答メッセージハンドラ名

メッセージハンドラ	説明
M.inbound_18X_INVITE	このメッセージハンドラは、180、181、182、および 183 を含むすべての着信 18X 応答に対して起動されます。
M.inbound_ANY_INVITE	このメッセージハンドラは、すべての 18X、2XX、3XX、4XX、5XX、および 6XX の応答を含む INVITE 要求のすべての着信応答に対して起動されます。
M.outbound_ANY_ANY	このメッセージハンドラは、要求メソッドに関係なく、すべての発信応答に対して起動されます。

メッセージハンドラを取得するルール

Cisco Unified CM では、次のルールを使用して、特定のメッセージのメッセージハンドラを見つけます。

1. メッセージハンドラでは、大文字と小文字が区別されます。
2. 方向は、着信または発信のいずれかです。
3. 方向は、常に、小文字で表記されます。
4. メッセージの方向は、SM に対する相対方向です。



(注) メッセージの方向は、SIP のダイアログの方向から完全に独立しています。

例

inbound_INVITE は有効なハンドラ名です。InBound_INVITE は有効なハンドラ名ではありません。

5. SIP メッセージから取得されるメソッド名は、スクリプトで適切なメッセージハンドラを見つける目的で、小文字に変換されます。
6. CUCM-SME では、最長一致基準を使用して、正しいメッセージハンドラが見つけます。

例

スクリプトに、inbound_ANY_ANY と inbound_183_INVITE の 2 つのメッセージハンドラがあるとします。Cisco Unified CM で 183 の応答を受信すると、これは最も明らかな一致のため、inbound_183_INVITE ハンドラが実行されます。



(注) 着信または発信は ANY (応答コード) および ANY (メソッド) でサポートされますが、特定の応答コードでの ANY (メソッド) ワイルドカードは、現在サポートされていません。

つまり、次のメッセージハンドラが有効です。

```
inbound_ANY_ANY
outbound_ANY_ANY
```

ただし、次のメッセージハンドラは無効です。

```
inbound_401_ANY
outbound_200_ANY
など
```