プロビジョニング スクリプト

プロビジョニング スクリプト

Cisco IP Phone は、オープンな公開された構文に基づくプロファイル形式を受け入れます。オープン形式では、XML に類似したシンプルな構文が使用されます。

このドキュメントの例では、オープン形式（XML スタイル）の構文による設定プロファイルが使用されます。付録 A「サンプル設定ファイル」でサンプル プロファイルを参照できます。

お客様の Cisco IP Phone の詳細については、お客様のデバイスのアドミニストレーション ガイドを参照してください。各ガイドでは、管理 Web サーバで設定できるパラメータについて説明しています。

設定プロファイルの形式

設定プロファイルは、Cisco IP Phone のパラメータ値を定義します。

設定プロファイルのオープン形式では、標準的な XML 作成ツールを使用して、パラメータと値をコンパイルします。

（注） UTF-8 文字セットのみがサポートされます。エディタでプロファイルを変更する場合、エンコーディング形式を変更しないでください。変更すると、Cisco IP Phone がファイルを認識できません。

Cisco IP Phone の各モデルの機能セットは異なっているため、パラメータ セットも異なります。

オープン形式（XML スタイル）プロファイル

オープン形式プロファイルは、要素を階層構造で記述する XML のような構文によるテキストファイルで、要素の属性と値を含んでいます。この形式により、標準的なツールを使用して設定ファイルを作成できるようになります。この形式の設定ファイルは、再同期操作の間に、プロビジョニング サーバから Cisco IP Phone に送信できます。このファイルは、バイナリ オブジェクトとしてコンパイルなしで送信できます。

Cisco IP Phone は、標準的なツールで生成される設定形式を受け入れることができます。この機能によって、既存のデータベースから設定プロファイルを生成するバックエンドのプロビジョニング サーバ ソフトウェアの開発が容易になります。

設定プロファイルの機密情報を保護するため、プロビジョニング サーバは、HTTPS が保護するセキュアなチャネルを使って、この種のファイルを電話に提供します。必要に応じて、gzip 圧縮アルゴリズム（RFC1951）を使用してファイルを圧縮できます。このファイルは、256 ビット AES 対称キー暗号化で暗号化できます。

例:オープン プロファイル形式

```
<device> <flat-profile>
<Resync_On_Reset> Yes
  </Resync_On_Reset>
<Resync_Periodic> 7200
  </Resync_Periodic>
<Profile_Rule>
    tftp://prov.telco.com:6900/cisco/config/CP_x8xx_3PCC.cfg
  </Profile_Rule>
</flat-profile> </device>
```

<flat-profile> 要素タグは、Cisco IP Phone の認識するすべてのパラメータの要素を囲みます。

(注)　2.0.6 以前のファームウェアバージョン を含む Cisco IP Phone は、オープン形式プロファイルをサポートしていません。

関連項目

- オープン プロファイル(XML スタイル)の圧縮と暗号化(2-5 ページ)

設定ファイルのコンポーネント

設定ファイルには、次のコンポーネントを含めることができます。

- 要素タグ
- 属性
- パラメータ
- 書式設定機能

要素タグのプロパティ

- Cisco IP Phone は、特別な <flat-profile> 要素でカプセル化された適切なパラメータ名で要素を認識しています。
- <flat-profile> 要素は、他の任意の要素内にカプセル化することができます。
- 要素名は、山カッコで囲まれています。
- ほとんどの要素名は、デバイスの管理 Web ページのフィールド名と同様ですが、以下の変更を伴います。
  - 要素名には、スペースや特殊文字が含まれないことがあります。管理 Web フィールド名から要素名を生成するには、すべてのスペースや特殊文字 [、]、(、)、/ をアンダースコアに置き換えます。

    例:<Resync_On_Reset> 要素は、[リセット時の再同期(Resync On Resetreset)] フィールドを表します。
  - 各要素名は固有である必要があります。管理 Web ページでは、同じフィールドは、[回線(Line)]、[ユーザ(User)]、および [拡張(Extension)] ページなど、複数の Web ページに表示される場合があります。ページ タブに表示される番号を示すには、要素名に [n] を追加します。

    例:<Dial_Plan[1]> 要素は、[回線 1(Line 1)] の [ダイヤル プラン(Dial Plan)] を表します。

- 始めの要素タグすべては、一致する終わりの要素タグを伴う必要があります。次に例を示します。

```
<device> <flat-profile>
<Resync_On_Reset> Yes
   </Resync_On_Reset>
<Resync_Periodic> 7200
   </Resync_Periodic>
<Profile_Rule>tftp://prov.telco.com: 6900/cisco/config/CP_x8xx_3PCC.cfg
   </Profile_Rule>
</flat-profile> </device>
```

- 要素タグは、大文字と小文字を区別します。
- 空の要素タグは許容されます。始めの要素タグを対応する要素タグなしで入力し、最後の山カッコ(>)の前にスペースとスラッシュを挿入します。この例では、プロファイル ルール B は空です。

```
<Profile_Rule_B />
```

- 認識されない要素名は無視されます。
- 再同期処理の際に、どのユーザ指定の値も上書きされないようにするため、空の要素タグを使用します。次の例では、ユーザの短縮ダイヤルの設定は変更されません。

```
<Speed_Dial_2_2_  ua="rw"/>
  <Speed_Dial_3_2_  ua="rw"/>
  <Speed_Dial_4_2_  ua="rw"/>
  <Speed_Dial_5_2_  ua="rw"/>
  <Speed_Dial_6_2_  ua="rw"/>
  <Speed_Dial_7_2_  ua="rw"/>
  <Speed_Dial_8_2_  ua="rw"/>
  <Speed_Dial_9_2_  ua="rw"/>
<device> </flat-profile> </device>
```

- 空の文字列に対応するパラメータを設定するため、空の値を使用します。始めと終わりの要素を、間に値を何も含めずに入力します。次の例では、GPP_A パラメータは空の文字列に設定されます。

```
<device> <flat-profile>
<GPP_A>
  </GPP_A>
</flat-profile> </device>
```

ユーザ アクセスの属性

ユーザ アクセスの **(ua)** 属性は、特定のパラメータに対するユーザ アカウントへのアクセスを制御します。**ua** 属性タグが要素タグで指定されなかった場合、工場出荷時のデフォルトのユーザ アクセスが、適用済みの対応するパラメータに対して適用されます。この属性は、管理者アカウントによるアクセスに影響しません。

ua 属性が存在する場合には、次のいずれかの値にする必要があります。

- na —アクセスなし
- ro —読み取りのみ
- rw —読み取りと書き込み

次の例は、**ua** 属性を示しています。

```
<device> <flat-profile>
  <SIP_TOS_DiffServ_Value_1_   ua="na"/>
  <Dial_Plan_1_   ua="ro"/>
  <Dial_Plan_2_   ua="rw"/>
</flat-profile> </device>
```

二重引用符で **ua** オプションの値を囲む必要があります。

LCD GUI のアクセス制御

<Phone-UI-User-Mode> パラメータを有効にすると、電話の GUI は、GUI がメニュー項目を表示するときに、関連するパラメータのユーザ アクセスの属性を受け入れます。

単一の設定パラメータに関連付けられたメニュー エントリの場合:

- “ua=na”（“ua” は「ユーザ アクセス（user access）」を意味する）属性を含むパラメータのプロビジョニングにより、エントリは非表示になります。
- “ua=ro” 属性を含むパラメータのプロビジョニングにより、エントリは読み取り専用で編集不可能になります。

複数の設定パラメータに関連付けられたメニュー エントリの場合:

- “ua=na” 属性を含むすべての関連するパラメータのプロビジョニングにより、エントリは非表示になります。

（注）　通常のユーザまたは管理者として LCD GUI からログインすると、すべての設定ページのデフォルトの表示は [ユーザ モード（User Mode）] になります。管理者ログインの後、このモードは [管理者モード（Admin Mode）] に切り替わり、属性は “ua=xx” となって、すべてのパラメータは無視されます。

パラメータのプロパティ

次のプロパティがパラメータに適用されます。

- プロファイルに指定されていないどのパラメータも、Cisco IP Phone で変更されることはありません。
- 認識されないパラメータは無視されます。
- Cisco IP Phone は、限られた数のパラメータ名に対する任意の設定可能なエイリアスを認識します。
- オープン形式プロファイルに同じパラメータ タグが複数回含まれている場合、最後のタグが先行するタグに上書きされます。パラメータの設定値の不注意な上書きを防ぐため、各プロファイルが指定するパラメータのインスタンスを最大でも 1 つにすることをお勧めします。

書式設定

次のプロパティが文字列の書式設定に適用されます。

- コメントは、標準的な XML 構文で作成できます。

```
<!-- My comment is typed here -->
```

- 先頭および後続のスペースは、読みやすさのために許容されますが、パラメータ値からは除外されます。
- 値の中での改行は、スペースに変換されます。
- <?　?> 形式の XML ヘッダーは許容されますが、Cisco IP Phone はこれを無視します。
- 特殊文字を入力するには、次の表に示すように、基本的な XML の文字エスケープを使用します。

特殊文字	XML のエスケープ シーケンス
&（アンパサンド）	&
<（より小さい）	<
>（より大きい）	>
’ (アポストロフィ)	'
”（二重引用符）	"

次の例では、文字エスケープは、ダイヤル プラン ルールに必要な記号よりも大きいことと小さいことを示すために入力されます。この例では、Dial_Plan[1] パラメータを（S0 <:18005551212>）に等しく設定する情報ホットラインのダイヤル プランを定義します。

```
<device> <flat-profile>
  <Dial_Plan_1_>
    (S0 &lt;:18005551212&gt;)
  </Dial_Plan_1_>
</flat-profile> </device>
```

- 10 進数および 16 進数の値（s.a.`(` と `.`）を使用する数字のエスケープが変換されます。
- ファームウェアは、完全な Unicode 文字セットをサポートしておらず、ASCII のサブセットのみをサポートしています。

オープン プロファイル（XML スタイル）の圧縮と暗号化

オープン設定プロファイルを圧縮して、プロビジョニング サーバのネットワーク負荷を軽減することができます。このプロファイルは、機密情報を保護するために暗号化することもできます。圧縮は必要ではありませんが、暗号化の前に行う必要があります。

オープン プロファイルの圧縮

サポートされている圧縮方法は、gzip 圧縮アルゴリズム（RFC1951）です。gzip ユーティリティと、同じアルゴリズム（zlib）を実装する圧縮ライブラリは、インターネット サイトから入手できます。

圧縮を識別するため、Cisco IP Phone は、gzip 互換のヘッダーを含めるための圧縮ファイルを要求します。元のオープン プロファイルで gzip ユーティリティを呼び出すと、ヘッダーが生成されます。Cisco IP Phone は、ダウンロードされたファイル ヘッダーを検査し、ファイル形式を確認します。

たとえば、`profile.xml` が有効なプロファイルの場合、`profile.xml.gz` も受け入れられます。次のコマンドのいずれも、このプロファイル タイプを生成できます。

- `>gzip profile.xml`

  元のファイルを圧縮ファイルと置き換えます。

- `>cat profile.xml | gzip > profile.xml.gz`

  元のファイルを残したまま、新しい圧縮ファイルを作成します。

圧縮のチュートリアルについては、「プロファイルの gzip 圧縮を開く」セクション（4-14 ページ）を参照してください。

AES の使用によるオープン プロファイルの暗号化

対称キー暗号化は、ファイルが圧縮されているかどうかにかかわりなく、オープン設定プロファイルの暗号化に使用できます。サポートされる暗号化アルゴリズムは、暗号ブロック連鎖モードで適用される、256 ビット キーを使用する American Encryption Standard（AES ）です。

（注）　圧縮および暗号化されたオープン形式プロファイルを Cisco IP Phone が認識できるようにするため、圧縮を暗号化に先行させる必要があります。「OpenSSL を使用したプロファイルの暗号化」セクション（4-15 ページ）は、暗号化に関するチュートリアルを提供しています。

OpenSSL 暗号化ツールは、さまざまなインターネット サイトからダウンロード可能で、暗号化を実行できます。256 ビット AES 暗号化のサポートには、AES コードを有効にするため、ツールの再コンパイルが必要になる場合があります。ファームウェアは、バージョン openssl-0.9.7c でテスト済みです。

暗号化ファイルについては、プロファイルは、次のコマンドによって生成されたものと同じ形式をもつファイルを要求します。

```
# example encryption key = SecretPhrase1234

openssl enc -e -aes-256-cbc -k SecretPhrase1234 -in profile.xml -out profile.cfg

# analogous invocation for a compressed xml file

openssl enc -e -aes-256-cbc -k SecretPhrase1234 -in profile.xml.gz -out profile.cfg
```

小文字の -k は、秘密キーに先行します。秘密キーは、いずれかのプレーン テキストの文字列で、ランダムな 64 ビット salt の生成に使用されます。-k 引数で指定された秘密を使用して、暗号化ツールは、ランダムな 128 ビット初期ベクトルと実際の 256 ビット暗号キーを生成します。

この形式の暗号化を設定プロファイルで使用する場合、ファイルを復号できるように、秘密キーの値を電話に知らせる必要があります。この値は、プロファイル URL で修飾子として指定されます。構文は次のとおりで、明示的な URL を使用します。

```
[--key "SecretPhrase1234"] http://prov.telco.com/path/profile.cfg
```

この値は、Profile_Rule パラメータのいずれかを使用してプログラムされます。このキーは、先立ってユニットに事前プロビジョニングしておく必要があります。秘密キーのブートストラップは、HTTPS を使用することで安全に実現できます。

対称キー暗号化によるオフラインでの設定プロファイルの事前暗号化により、再同期プロファイルに HTTP を使用することが可能になります。プロビジョニング サーバは、Cisco IP Phone 導入後の最初のプロビジョニングを処理するために HTTPS を使用します。この機能は、大規模導入時の HTTPS サーバの負荷を軽減します。

最終的なファイル名は特定の形式を必要としませんが、通常、.cfg 拡張子で終わるファイル名は設定プロファイルを示します。

コメント

開発とスクリプト作成の際、パラメータ値の先頭に # の文字を入力して、プロビジョニング パラメータを一時的に無効にすると便利です。これにより、パラメータの残りのテキストが効果的にコメント アウトされます。

例

値 `"# http://192.168.1.200/sample.cfg"` を含む Profile_Rule は、空の Profile_Rule に相当します。# 文字によるコメント機能は、Profile_Rule*、Upgrade_Rule、および Resync_Trigger_* パラメータに適用されます。

マクロ展開

複数のプロビジョニング パラメータは、評価される前に内部でのマクロ展開により処理されます。この評価前の手順によって、Cisco IP Phone の再同期およびアップグレード アクティビティの制御がより柔軟になります。

次のパラメータ グループは、評価の前にマクロ展開により処理されます。

- Resync_Trigger_*
- Profile_Rule*
- Log_xxx_Msg
- Upgrade_Rule

特定の条件下では、一部の汎用パラメータ（GPP_*）も、[オプション再同期引数（Optional Resync Arguments）] セクションで明示的に示されているように、マクロ展開により処理されます。

マクロ展開の際、名前付き変数の内容は、$NAME および $(NAME) 形式の式を置き換えます。そのような変数には、汎用パラメータ、複数の製品識別子、特定のイベント タイマー、プロビジョニングの状態の値が含まれます。完全なリストについては、「マクロ展開変数」セクション（5-5 ページ）を参照してください。

次の例では、式 $(MAU) を使用して MAC アドレス 000E08012345 を挿入します。

管理者は次の式を入力します。$(MAU)config.cfg
MAC アドレス 000E08012345 のデバイスのマクロ展開の結果は次のとおりです。
000E08012345config.cfg

マクロ名が認識されない場合には、展開されません。たとえば、STRANGE は有効なマクロ名として認識されませんが、MAU は有効なマクロ名として認識されます。

管理者は次の式を入力します。$STRANGE$MAU.cfg
MAC アドレス 000E08012345 のデバイスのマクロ展開の結果は次のとおりです。
$STRANGE000E08012345.cfg

マクロ展開は、再帰的に適用されません。たとえば、$$MAU" は $MAU" に展開し（$$ が展開される）、MAC アドレスになることはありません。

特殊な用途のパラメータである GPP_SA から GPP_SD の内容は、マクロ式 $SA から $SD にマッピングされます。これらのパラメータは、再同期 URL の **--key** オプションの引数としてのみマクロ展開します。

マクロ式は、MAC アドレスの一部分と同様、マクロ変数の部分文字列のみを使用するように、式を修飾できます。

部分文字列マクロ展開の構文は、$(NAME:p) と $(NAME:p:q) で、p と q は負ではない整数です。この展開により、文字のオフセット p で開始される長さ q（q が指定されない場合には文字列の終端まで）のマクロ変数部分文字列が生成されます。次の例を参照してください。

管理者は次の式を入力します。$(MAU:4)
MAC アドレス 000E08012345 のデバイスのマクロ展開の結果は次のとおりです。08012345

管理者は次の式を入力します。$(MAU:8:2)
MAC アドレス 000E08012345 のデバイスのマクロ展開の結果は次のとおりです。23

条件式

条件式で、再同期イベントをトリガーし、再同期およびアップグレードの操作に対する代替 URL を選択できます。

条件式は、比較のリストで構成されており、**and** 演算子によって分割されます。すべての比較は、条件が true になる要件を満たしている必要があります。

各比較は、3 つのタイプのリテラルのうち 1 つを関連付けることができます。

- 整数値
- ソフトウェアまたはハードウェアのバージョン番号
- 二重引用符の文字列

バージョン番号

3PCC 電話の公式リリース ソフトウェア バージョンは、7800 シリーズの電話の 形式 sip78xx.v1-v2-v3-v4-3PCC と、8800 シリーズの電話の sip88xx.v1-v2-v3-v4-3PCC 形式を使用しています。比較文字列には、同じ形式を使用する必要があります。そうしない場合、結果として形式解析エラーが発生します。

このソフトウェア バージョンでは、v1-v2-v3-v4 には異なる桁と文字を指定できますが、数字で始まっている必要があります。ソフトウェア バージョンを比較する際、v1-v2-v3-v4 は順に比較され、左端の桁が後続の桁よりも優先されます。

v[x] に数字のみが含まれている場合、その数字が比較されます。v[x] に数字とアルファベット文字が含まれている場合、まず数字が比較され、次に文字がアルファベット順に比較されます。

有効なバージョン番号の例

sip78yy.10-3-1-7-3PCC

一方、10.3.1 は、無効な形式です。

比較

sip88xx.10-3-1-7-3PCC > sip88xx.9-3-1-7-3PCC

sip78xx.10-3-1-7-3PCC < sip78xx.10-3-1MN-1-3PCC

囲まれた文字列は、等しいか等しくないかについて比較できます。整数とバージョン番号も、算術的に比較できます。比較演算子は、記号または略語で表すことができます。略語は、オープン形式プロファイルで条件を表すのに便利です。

演算子	代替構文	説明	整数とバージョンのオペランドに適用可能	囲まれた文字列のオペランドに適用可能
=	eq	次の値と等しい	○	○
!=	ne	等しくない	○	○
<	lt	より少ない	○	×
<=	le	右辺と比較して小さいか等しい	○	×
>	gt	より大きい	○	×
>=	ge	右辺と比較して大きいか等しい	○	×

2.0.6 よりも前のファームウェアバージョンへのレガシー サポートの場合、not-equal-to 演算子を、（2 文字の != 文字列の代わりに）単一の！の文字で表すことができます。

条件式には、通常、マクロ展開された変数が含まれます。次に例を示します。

```
$REGTMR1 gt 300 and $PRVTMR gt 1200 and "$EXTIP" ne ""

$SWVER ge 2.0.6 and "$CCERT" eq "Installed"
```

文字列リテラルが要求されるときに、マクロ変数を二重引用符で囲むことは重要です。バージョン番号の番号が要求されるときには、そうしないでください。

2.0.6 より前のファームウェア バージョンのレガシー サポートの場合、左辺オペランドのない関係式は、暗黙的な左辺として $SWVER を使用します。たとえば、! 1.0.33 は、$SWVER = 1.0.33 に相当します。

Profile_Rule* Upgrade_Rule パラメータのコンテキストで使用すると、次のアップグレード ルールの例に示すように、条件式を構文 "(expr)?" で囲む必要があります。

```
($SWVER ne sip78xx.10-3-1-10-3PCC)? http://ps.tell.com/sw/sip78xx.10-3-1-10-3PCC.loads
```

Resync_Trigger_* パラメータを設定する場合には、カッコを含む上記の構文を使用しないでください。

割り当て式

任意のパラメータが、Profile_Rule* と Upgrade_Rule パラメータのコンテキストで事前に割り当てられた値である場合もあります。この手順によって、プロファイルを取得する前に、割り当ての実行が発生します。

このような割り当ての構文は、個々のパラメータ割り当てのリストで、(assignments)! のようにカッコで囲まれており、各割り当ては次の形式で表されます。

ParameterXMLName = "Value"

認識されたパラメータ名は、XML ベースのプロファイル名に対応します。

いずれのパラメータにも、この方法で新しい値を割り当てる事が可能で、マクロ展開が適用されます。たとえば、次の例は有効な割り当て式です。

```
(User_ID_1_ = "uid$B" ; GPP_C = "" ; GPP_D = "$MA" ;)!
```

単一の小文字 a から p は、汎用パラメータ GPP_A から GPP_P を参照することもできます。上の例は、次の例と同等です。

```
(User_ID_1_ = "uid$B" ; c = "" ; d = "$MA" ;)!
```

読みやすいようにスペースを使用します。

URL 構文

標準的な URL 構文を使用して、設定ファイルとファームウェア ロードを、各々 Profile_Rule* および Upgrade_Rule パラメータに取得する方法を指定します。構文は次のようになります。

```
[ scheme:// ] [ server [:port]] filepath
```

`scheme` は次のいずれかの値です。

- tftp
- http
- https

`scheme` を省略すると、tftp が使用されます。server には、DNS に認識されるホスト名または数値による IP アドレスを使用できます。port は、接続先 UDP または TCP ポート番号です。filepath は、ルート ディレクトリ (/) から始める必要があります。これは絶対パスである必要があるためです。

`server` が見つからない場合、DHCP（オプション 66）で指定される tftp サーバが使用されます。

`port` が見つからない場合、指定されたスキームの標準ポートが使用されます。（tftp は UDP ポート 69、http は TCP ポート 80、https は TCP ポート 443を使用します）。

filepath が存在する必要があります。静的ファイルを参照する必要はありませんが、CGI で取得されたダイナミック コンテンツを指定することもできます。

マクロ展開は、URL 内で適用されます。有効な URL の例を次に示します。

```
/$MA.cfg
/cisco/sip78xx.10-3-1-3PCC.loads
192.168.1.130/profiles/init.cfg
tftp://prov.call.com/cpe/cisco$MA.cfg
http://neptune.speak.net:8080/prov/$D/$E.cfg
https://secure.me.com/profile?Linksys
```

オプション再同期引数

オプションの引数である **key** が、角カッコで集合的に囲まれて、Profile_Rule* パラメータに入力した URL の前に付く場合があります。

key

Key オプションは、暗号キーの指定に使用されます。明示的なキーによる、暗号化されたプロファイルの復号化が必要です。キー自体は、**--key** の語に続く（おそらく引用された）文字列として指定されます。

使用例

```
[--key VerySecretValue]
[--key "my secret phrase"]
[--key a37d2fb9055c1d04883a0745eb0917a4]
```

カッコで囲まれたオプションの引数は、マクロ展開されます。特殊な用途のパラメータ（GPP_SA から GPP_SD）は、Key オプションの引数として使用される場合に限り、マクロ変数にマクロ展開されます。次の例を参照してください。

```
[--key $SC]
[--key "$SD"]
```

オープン形式プロファイルの場合、**--key** に対する引数は、**openssl** に与えられた **-k** オプションと一致している必要があります。

IP テレフォニー デバイスへのプロファイルの適用

XML 設定スクリプトを作成した後、Cisco IP Phone に渡して適用する必要があります。設定を適用するには、TFTP サーバから、または cURL を使用するかのいずれかの方法で、設定ファイルを IP 電話にポストします。

TFTP サーバから IP 電話への設定ファイルのポスト

設定ファイルを PC の TFTP サーバ アプリケーションにポストするには、次の手順を実行します。

ステップ 1　PC を電話の LAN に接続します。

ステップ 2　PC の TFTP サーバ アプリケーションを開始し、設定ファイルが TFTP サーバのルート ディレクトリで使用できることを確認します。

ステップ 3　Web ブラウザで、Cisco IP Phone の LAN IP アドレス、コンピュータの IP アドレス、ファイル名、およびログイン クレデンシャルを入力します。次の形式を使用します。

http://<*WAN_IP_Address*>/admin/resync?tftp://<*PC_IP_Address*>/<*file_name*>&xuser=admin&xpassword=<*password*>

例：

```
http://192.168.15.1/admin/resync?tftp://192.168.15.100/my_config.xml&xuser=admin&xpassword
=admin
```

cURL を使用した IP 電話への設定ファイルのポスト

cURL を使用して設定を Cisco IP Phone にポストするには、次の手順を実行します。このコマンド ライン ツールは、URL の構文でデータを転送するために使用されます。cURL をダウンロードする方法については、次を参照してください:

http://curl.haxx.se/download.html

ステップ 1　Cisco IP Phone の LAN ポートに PC を接続します。

ステップ 2　次の cURL コマンドを入力して、設定ファイルを Cisco IP Phone にポストしてください。

```
curl -d @my_config.xml
"http://192.168.15.1/admin/config.xml&xuser=admin&xpassword=admin"
```

プロビジョニング パラメータ

このセクションでは、機能に応じて大まかにまとめられたプロビジョニング パラメータについて説明します。

次のプロビジョニング パラメータのタイプが存在します。

- 汎用
- イネーブル
- トリガー
- 設定可能なスケジュール
- プロファイル ルール
- アップグレード ルール

汎用パラメータ

汎用パラメータ GPP_* は、特定のプロビジョニング サーバ ソリューションと連携するように Cisco IP Phone を設定するときに、自由文字列のレジスタとして使用されます。GPP_* パラメータは、デフォルトでは空です。これらは、次を含むさまざまな値に設定できます。

- 暗号化キー
- URL
- マルチステージ プロビジョニング ステータス情報
- Post 要求テンプレート
- パラメータ名エイリアス マップ
- 最終的に完全なパラメータ値に組み込まれる部分文字列値。

GPP_* パラメータは、他のプロビジョニング パラメータ内でのマクロ展開に利用できます。この目的のため、GPP_A から GPP_P の内容を識別するには、単一の小文字マクロ名（A から P）があれば十分です。また、2 文字の大文字のマクロ名 SA から SD は、**key** URL オプションの引数として使用される特殊なケースとして、GPP_SA から GPP_SD を識別します。

これらのパラメータは、プロビジョニングとアップグレードのルールで変数として使用できます。$GPP_A など、変数名の前に ‘$’ の文字を付けることにより参照されます。

汎用パラメータの使用

たとえば、GPP_A に 文字列 ABC が含まれ、GPP_B に 123 が含まれる場合、式 AB マクロ は ABC123 に展開します。

ステップ 1　電話の Web ユーザ インターフェイス で、[管理者ログイン（Admin Login）] > [詳細（advanced）] > [音声（Voice）] > [プロビジョニング（Provisioning）] と移動します。

ステップ 2　[汎用パラメータ（General Purpose Parameters）] セクションまでスクロールします。

ステップ 3　フィールドに有効な値、GPP A から GPP P を入力します。

ステップ 4　[すべての変更を送信（Submit All Changes）] をクリックします。

有効化

Provision_Enable および Upgrade_Enable パラメータは、プロファイルの再同期とファームウェア アップグレードの操作すべてを制御します。これらのパラメータは、再同期とアップグレードをそれぞれ個別に制御します。これらのパラメータは、管理 Web サーバによって発行される URL コマンドの再同期とアップグレードも制御します。両方のパラメータは、デフォルトでは [はい（yes）] に設定されています。

Resync_From_SIP パラメータは、再同期操作を要求します。SIP NOTIFY イベントは、サービス プロバイダーのプロキシ サーバから Cisco IP Phone へ送信されます。有効にすると、プロキシが再同期を要求できるようになります。これを実行するため、プロキシは、Event: resync ヘッダーを含む SIP NOTIFY メッセージをデバイスに送信します。

デバイスは、401 応答（使用したクレデンシャルを理由に認証を拒否）でその要求にチャレンジします。デバイスは、プロキシからの再同期要求を引き受ける前に、認証済みの後続要求を求めます。Event: reboot_now および Event: restart_now ヘッダーは、それぞれコールド再起動とウォーム再起動を実行し、これらもチャレンジを受けます。

残り 2 つのイネーブルは、Resync_On_Reset と Resync_After_Upgrade_Attempt です。これらのパラメータは、電源投入ソフトウェアの再起動と各アップグレード試行の後に、デバイスが再同期操作を実行したかどうかを判定します。

Resync_On_Reset を有効にすると、デバイスは、リセットが実行される前のブートアップ シーケンスに先立って、ランダム遅延を発生させます。この遅延は、Resync_Random_Delay（秒単位）が指定する値を上限とするランダムな時間です。同時に起動する電話のプールの場合、この遅延が、各ユニットからの再同期要求の開始時刻を分散させます。この機能は、地域の停電時に、大規模な宅内導入で役立つ場合があります。

トリガー

Cisco IP Phone は、特定の間隔で、または特定の時間に再同期することを可能にします。

特定の間隔での再同期

Cisco IP Phone は、プロビジョニング サーバと定期的に再同期されるよう設計されています。再同期の間隔は、Resync_Periodic（秒単位）で設定されます。この値が空の場合、デバイスは定期的に再同期されません。

再同期は、通常、音声回線がアイドル状態になっているときに発生します。音声回線がアクティブで、再同期が予定されている場合、Cisco IP Phone は、回線が再度アイドル状態になるまで再同期手順を延期します。ただし、電話が Forced_Resync_Delay（秒単位）を超えて待機することはありません。再同期によって、設定パラメータ値が変更される場合があります。この変更により、ファームウェアのリブートが発生し、再同期時にアクティブだった音声接続が切断されます。

Cisco IP Phone がサーバからプロファイルを取得できなかった場合、ダウンロードしたファイルが破損していた場合、または内部エラーが発生した場合には、再同期操作が失敗する可能性があります。デバイスは、Resync_Error_Retry_Delay（秒単位）で指定された時間が経過した後、再び再同期を試行します。Resync_Error_Retry_Delay が 0 に設定されている場合、再同期の試行が失敗した後に、デバイスが再同期を試みることはありません。

アップグレードが失敗すると、Upgrade_Error_Retry_Delay（秒単位）の後に再試行が実行されます。

2 つの設定可能なパラメータ、Resync_Trigger_1 と Resync_Trigger_2 を使用して、再同期を条件付きでトリガーできます。各パラメータは、マクロ展開される条件式でプログラムできます。これらのパラメータのいずれかの条件が true と評価される場合、再同期操作は、定期的な再同期タイマーの期限が切れた場合のようにトリガーされます。

次の例の条件は、再同期をトリガーします。この例では、電話の最後のアップグレード試行から 5 分（300秒）以上が経過し、最後の再同期試行から少なくとも 10 分（600秒）が経過しています。

```
$UPGTMR gt 300 and $PRVTMR ge 600
```

特定の時間での再同期

Resync_At パラメータは、特定の時間に電話が再同期されることを可能にします。このパラメータは、24 時間形式（hhmm）を使用して時間を指定します。

Resync_At_Random_Delay パラメータは、指定されていない遅延時間で電話が再同期されることを可能にします。このパラメータは、正の整数の形式を使用して時間を指定します。

同時に再同期するよう設定された多数の電話から、サーバに再同期要求が押し寄せることは避ける必要があります。そのため、電話は、指定された時間の最大で 10 分後に再同期をトリガーします。

たとえば、再同期時間を 1000（午前 10 時）に設定すると、電話は、午前 10 時と午前 10 時 10 分の間のいずれかの時間に再同期をトリガーします。

デフォルトでは、この機能は無効になっています。Resync_At パラメータがプロビジョニングされると、Resync_Periodic パラメータは無視されます。

設定可能なスケジュール

次のプロビジョニング パラメータを使用して、定期的な再同期のスケジュールを設定することが可能で、再同期およびアップグレードが失敗した場合の再試行間隔も指定できます。

- Resync_Periodic
- Resync_Error_Retry_Delay
- Upgrade_Error_Retry_Delay

各パラメータは、単一の遅延値（秒単位）を受け入れます。新しく展開された構文は、連続的な遅延要素のカンマ区切りリストを許容します。シーケンスの最後の要素は、暗黙的に際限なく繰り返されます。次に例を示します。

```
Resync_Periodic=7200
Resync_Error_Retry_Delay=1800,3600,7200,14400
```

前の例では、Cisco IP Phone が 2 時間ごとに定期的に再同期されます。再同期障害が発生すると、デバイスは、30 分、1 時間、2 時間、4 時間の間隔で再試行します。正常に再同期されるまで、デバイスは、4 時間間隔で試行し続けます。

次の例のとおり、必要に応じて、プラス記号を使用して、ランダムな追加の遅延を付加する別の数値を指定することもできます。

```
Resync_Periodic=3600+600
Resync_Error_Retry_Delay=1800+300,3600+600,7200+900
```

前の例で、デバイスは、1 時間（プラス最大 10 分の追加のランダム遅延）ごとに定期的に再同期されます。再同期障害が発生すると、デバイスは、次の間隔で再試行します。30 分（プラス 最大 5 分）。1 時間（プラス 最大 10 分）、2 時間（プラス最大 15 分）。正常に再同期されるまで、デバイスは、2 時間間隔（プラス最大 15 分）で試行し続けます。

別の例を示します。

```
Upgrade_Error_Retry_Delay  =  1800,3600,7200,14400+3600
```

この例では、リモート アップグレードに失敗した場合、デバイスは、30 分後にアップグレードを再試行し、次に 1 時間後、次は 2 時間後に再試行します。それでもアップグレードに失敗する場合、デバイスは、アップグレードが成功するまで、4 時間から 5 時間ごとに再試行します。

プロファイル ルール

Cisco IP Phone は、複数のリモート設定プロファイル パラメータ（Profile_Rule*）を提供します。そのため、各再同期操作は、異なるサーバが管理する複数のファイルを取得できます。

最も簡単なシナリオでは、デバイスは、関係するすべての内部パラメータを更新する、中央サーバの単一のプロファイルに対して定期的に再同期されます。そうでない場合、プロファイルを異なるファイルの間で分割することができます。1 つのファイルは、導入時の Cisco IP Phone すべてに対して共通のファイルになります。他とは異なる固有のファイルが各アカウントに提供されます。暗号キーと証明書情報は、別のサーバに保存されている、さらに別のプロファイルから取得することも可能です。

再同期操作の時間になると、Cisco IP Phone は、4 つの Profile_Rule* パラメータを順に評価します。

1. Profile_Rule
2. Profile_Rule_B
3. Profile_Rule_C
4. Profile_Rule_D

各評価の結果、リモート プロビジョニング サーバからプロファイルが取得され、いくつかの内部パラメータのアップデートが発生する可能性があります。評価が失敗すると、再同期シーケンスは中断され、Resync_Error_Retry_Delay パラメータ（秒単位）によって指定された開始時間から再試行されます。すべての評価が成功すると、デバイスは、Resync_Periodic パラメータで指定された秒数の間待機した後、次の再同期を実行します。

各 Profile_Rule* パラメータの内容は、一連の選択肢で構成されます。それらの選択肢は、|（パイプ）文字で区切られます。各選択肢は、条件式、割り当て式、プロファイルの URL、および関連 URL のオプションで構成されています。これらすべてのコンポーネントは、各選択肢内のオプションです。次に示すのは、オプションの有効な組み合わせと、それらが存在する場合に従わなければならない表示順序です。

```
[ conditional-expr ] [ assignment-expr ] [[ options ] URL ]
```

各 Profile_Rule* パラメータ内で、最後の 1 つを除くすべての選択肢は、条件式を示す必要があります。この式は、次のように評価され処理されます。

1. 条件は、true と評価されるものが見つかるまで（または条件式を含まない 1 つの選択肢が見つかるまで）、左から右に評価されます。
2. いずれかの割り当て式を伴う場合には、それも評価されます。
3. 選択肢の一部として URL が指定される場合、指定された URL にあるプロファイルのダウンロードが試行されます。システムは、内部パラメータの更新を状況に応じて試行します。

すべての選択肢が条件式を含むものの、いずれも true と評価されない場合（またはプロファイル ルール全体が空の場合）、Profile_Rule* パラメータの全体がスキップされます。シーケンス内の次のプロファイル ルール パラメータが評価されます。

単一の Profile_Rule* パラメータに対して有効なプログラミングの例

この例は、指定された URL のプロファイルに対して無条件で再同期し、リモート プロビジョニング サーバに対して HTTP GET 要求を実行します。

```
http://remote.server.com/cisco/$MA.cfg
```

この例では、デバイスは、[回線 1（Line 1）] の登録ステータスに応じて、2 つの異なる URL に対して再同期します。登録が失われた場合、デバイスは、CGI スクリプトに対して HTTP POST を実行します。デバイスは、デバイスの状態に関する追加情報を提供する可能性のある、マクロ展開された GPP_A の内容を送信します。

```
($PRVTMR ge 600)? http://p.tel.com/has-reg.cfg
| [--post a] http://p.tel.com/lost-reg?
```

この例では、デバイスは、同じサーバに対して再同期されます。デバイスは、ユニットに証明書がインストールされていない場合、追加情報を提供します（2.0 よりも前のレガシー ユニットの場合）。

```
("$CCERT" eq "Installed")? https://p.tel.com/config?
| https://p.tel.com/config?cisco$MAU
```

この例では、[回線 1（Line 1）]は、GPP_A が最初の URL で Provisioned に等しくなるよう設定されるまでは無効です。その後、第 2 の URL に対して再同期されます。

```
("$A" ne "Provisioned")? (Line_Enable_1_ = "No";)! https://p.tel.com/init-prov
| https://p.tel.com/configs
```

この例では、サーバが返すプロファイルは、XML 要素タグを含むと想定します。これらのタグは、GPP_B に保存されるエイリアス マップにより、適切なパラメータ名に再配置される必要があります。

```
[--alias b] https://p.tel.com/account/$PN$MA.xml
```

再同期は、通常、要求されたプロファイルがサーバから受信されなかった場合に失敗と見なされます。Resync_Fails_On_FNF パラメータは、このデフォルトの動作をオーバーライドできます。Resync_Fails_On_FNF が [いいえ(No)] に設定されると、デバイスは、サーバからの file-not-found 応答を正常な再同期として受け入れます。Resync_Fails_On_FNF のデフォルト値ははいです。

アップグレード ルール

Cisco IP Phone は、1 つの設定可能なリモート アップグレード パラメータ、Upgrade_Rule を提供します。このパラメータは、プロファイル ルール パラメータと類似した構文を受け入れます。URL オプションは、アップグレードではサポートされませんが、条件式と割り当て式は使用できます。条件式を使用すると、| 文字で区切られた複数の選択肢をパラメータに含めることができます。各選択肢の構文は次のとおりです。

```
[ conditional-expr ] [ assignment-expr ] URL
```

Profile_Rule* パラメータの場合、Upgrade_Rule パラメータは、条件式の要件が満たされるか、選択肢が条件式を含まなくなるまで、各選択肢を評価します。付属の割り当て式が指定されている場合、それも評価されます。次に、指定された URL に対するアップグレードが試行されます。

Upgrade_Rule が条件式を含まない URL を含む場合、デバイスは、URL の指定するファームウェア イメージに対してアップグレードを実行します。マクロ展開とルールの評価の後、デバイスは、ルールが変更されるか、scheme + server + port + filepath の有効な組み合わせが変更されるまで、アップグレードを再試行しません。

ファームウェア アップグレードを試行する際、デバイスは、手順の最初に音声を無効にし、手順の最後にリブートします。デバイスは、音声回線が現在非アクティブになっている場合にのみ、Upgrade_Rule の内容に基づくアップグレードを自動的に開始します。

Cisco IP Phone 7800 シリーズ の場合を次の例に示します。

```
http://p.tel.com/firmware/sip78xx.10-3-1-3PCC.loads
```

Cisco IP Phone 8800 シリーズ の場合は次のとおりです。

```
http://p.tel.com/firmware/sip88xx.10-3-1-3PCC.loads
```

この例では、Upgrade_Rule は、指定された URL に保存されたイメージに対してファームウェアをアップグレードします。

Cisco IP Phone 7800シリーズの別の例を次に示します。

```
("$F" ne "beta-customer")? http://p.tel.com/firmware/sip78xx.10-3-1-3PCC.loads
| http://p.tel.com/firmware/sip78xx.10-3-1-3PCC.loads
```

Cisco IP Phone 8880 シリーズ の場合は次のとおりです。

```
("$F" ne "beta-customer")? http://p.tel.com/firmware/sip88xx.10-3-1-3PCC.loads
| http://p.tel.com/firmware/sip88xx.10-3-1-3PCC.loads
```

この例では、汎用パラメータ GPP_F の内容に基づいて、2 つのイメージのいずれかをロードするようユニットに指示します。

デバイスは、便利なカスタマイズ オプションとなり得る、ファームウェア リビジョン番号に関するダウングレード制限を設定できます。有効なファームウェア リビジョン番号が Downgrade_Rev_Limit パラメータで設定されると、デバイスは、指定された制限よりも前のファームウェア バージョンに対するアップグレードの試行を拒否します。

データ型

設定プロファイル パラメータで使用されるのは、次のデータ型です。

- Uns<n> —符号なし n ビット値（n = 8、16、または32）。値が n ビットにフィットする限り、10 進数または 16 進数の形式（たとえば 12 または 0x18）で指定できます。
- Sig<n> —符号付 n ビット値。10 進数または 16 進数の形式で指定できます。“-” 記号を負の値の前に付ける必要があります。正の値の前 + 記号はオプションです。
- Str<n> —最大 n 個の非予約文字を含む一般的な文字列。
- Float<n> —小数点以下第 n 位までを含む浮動小数点値。
- Time<n> —小数点以下第 n 位までを含む秒単位の継続時間。追加で指定された小数点以下の桁は無視されます。
- PwrLevel —小数点以下第 1 位を含む、dBm で表される電力レベル。–13.5 や 1.5（dBm）など。
- Bool — “yes” または “no” のいずれかのブール値。
- {a,b,c,…} — a、b、c、…からの選択肢。
- IP — x.x.x.x の形式の IP アドレス。x は 0 と 255 の間。例：10.1.2.100
- Port — TCP/UDP ポート番号（0-65535）。10 進数または 16 進数の形式で指定できます。
- UserID — URL に表示されるユーザ ID。最大 63 文字。
- FQDN —完全修飾ドメイン名。“sip.Cisco.com:5060” または “109.12.14.12:12345” など。最大 63 文字を指定できます。
- Phone —電話番号の文字列。14081234567、*69、*72、345678など。または、1234@10.10.10.100:5068 や jsmith@Cisco.com などの一般的な URL。この文字列には最大 39 文字を含めることができます。
- ActCode —補足サービスのアクティベーション コード。*69 など。このコードには最大 7 文字を含めることができます。
- PhTmplt —電話番号のテンプレート。各テンプレートには、カンマ（,）で区切られる 1 つ以上のパターンを含めることができます。各パターンの冒頭のスペースは無視されます。“?” と “*” はワイルドカード文字を示します。正確に表わすには、%xx を使用します。たとえば、%2a は * を表します。このテンプレートには最大 39 文字を含めることができます。例：“1408*, 1510*”、“1408123????, 555?1.”
- RscTmplt — SIP 応答ステータス コードのテンプレート。“404, 5*”、“61?”、“407, 408, 487, 481” など。最大 39 文字を指定できます。
- CadScript —信号のパターン パラメータを指定する小スクリプト。最大 127 文字。

  構文は $S_1[;S_2]$ で、次の意味があります。

  $S_i=D_i(on_{i,1}/off_{i,1}[,on_{i,2}/off_{i,2}[,on_{i,3}/off_{i,3}[,on_{i,4}/off_{i,4}[,on_{i,5}/off_{i,5}[,on_{i,6}/off_{i,6}]]]]])$ で、*セクション（section）*として知られています。$on_{i,j}$ と $off_{i,j}$ は、*セグメント（segment）*の秒単位の on/off 継続時間です。i = 1 または 2、および j = 1 から 6です。D_i は、セクションの継続時間の合計（秒単位）です。すべての継続時間には、1 ms 単位の精度を実現するため、小数点以下第 3 位まで含めることができます。ワイルドカード文字 “*” は無限の期間を意味します。セクション内のセグメントは、順に実行され、全継続期間が実行されるまで繰り返されます。

例 1:

```
60(2/4)

Number of Cadence Sections = 1
Cadence Section 1: Section Length = 60 s
Number of Segments = 1
Segment 1: On=2s, Off=4s

Total Ring Length = 60s
```

例 2 —特殊呼び出し音（短、短、短、長）

```
60(.2/.2,.2/.2,.2/.2,1/4)

Number of Cadence Sections = 1
Cadence Section 1: Section Length = 60s
Number of Segments = 4
Segment 1: On=0.2s, Off=0.2s
Segment 2: On=0.2s, Off=0.2s
Segment 3: On=0.2s, Off=0.2s
Segment 4: On=1.0s, Off=4.0s

Total Ring Length = 60s
```

- FreqScript —呼び出し音の周波数およびレベルのパラメータを指定する小スクリプト。127 文字まで含めることができます。構文は $F_1@L_1[,F_2@_{L2}[,F_3@L_3[,F_4@L_4[,F_5@L_5[,F_6@L_6]]]]]$ で、F_1–F_6 は Hz 単位の周波数を表します（符号なし整数のみ）。L_1–L_6 は、対応する dBm 単位のレベルを表します（小数点以下第 1 位までを含む）。カンマ前後のスペースは許容されますが、推奨されません。

  例 1 —コール ウェイティング トーン

```
440@-10

Number of Frequencies = 1
Frequency 2 = 440 Hz at -10 dBm
```

  例 2 —ダイヤル トーン

```
350@-19,440@-19

Number of Frequencies = 2
Frequency 1 = 350 Hz at -19 dBm
Frequency 2 = 440 Hz at -19 dBm
```

- ToneScript —コール プログレス トーンの周波数、レベル、パターンのパラメータを指定する小スクリプト。スクリプトには最大 127 文字を指定できます。構文は FreqScript;Z_1[;Z_2] です。セクション Z_1 は、CadScript の S_1 に類似していますが、各 on/off セグメントの後に、周波数コンポーネント パラメータ $Z_1 = D_1(on_{i,1}/off_{i,1}/f_{i,1}[,on_{i,2}/off_{i,2}/f_{i,2}\ [,on_{i,3}/off_{i,3}/f_{i,3}\ [,on_{i,4}/off_{i,4}/f_{i,4}\ [,on_{i,5}/off_{i,5}/f_{i,5}\ [,on_{i,6}/off_{i,6}/f_{i,6}]]]]])$ が続きます。$fi,j = n_1[+n_2]+n_3[+n_4[+n_5[+n_6]]]]$ です。$1 < n_k < 6$ は、このセグメントで使用される、FreqScript の周波数コンポーネントを指定します。複数の周波数コンポーネントが 1 つのセグメントで使用される場合、それらのコンポーネントは 1 つにまとめられます。

  例 1 —ダイヤル トーン

```
350@-19,440@-19;10(*/0/1+2)

Number of Frequencies = 2
Frequency 1 = 350 Hz at -19 dBm
Frequency 2 = 440 Hz at -19 dBm
Number of Cadence Sections = 1
Cadence Section 1: Section Length = 10 s
```

```
Number of Segments = 1
Segment 1: On=forever, with Frequencies 1 and 2

Total Tone Length = 10s
```

例 2 —断続トーン

```
350@-19,440@-19;2(.1/.1/1+2);10(*/0/1+2)

Number of Frequencies = 2
Frequency 1 = 350 Hz at -19 dBm
Frequency 2 = 440 Hz at -19 dBm
Number of Cadence Sections = 2
Cadence Section 1: Section Length = 2s
Number of Segments = 1
Segment 1: On=0.1s, Off=0.1s with Frequencies 1 and 2
Cadence Section 2: Section Length = 10s
Number of Segments = 1
Segment 1: On=forever, with Frequencies 1 and 2

Total Tone Length = 12s
```

例 3 — SIT トーン

```
985@-16,1428@-16,1777@-16;20(.380/0/1,.380/0/2,.380/0/3,0/4/0)
Number of Frequencies = 3
Frequency 1 = 985 Hz at -16 dBm
Frequency 2 = 1428 Hz at -16 dBm
Frequency 3 = 1777 Hz at -16 dBm
Number of Cadence Sections = 1
Cadence Section 1: Section Length = 20s
Number of Segments = 4
Segment 1: On=0.38s, Off=0s, with Frequency 1
Segment 2: On=0.38s, Off=0s, with Frequency 2
Segment 3: On=0.38s, Off=0s, with Frequency 3
Segment 4: On=0s, Off=4s, with no frequency components
Total Tone Length = 20s
```

- ProvisioningRuleSyntax —設定再同期およびファームウェア アップグレード ルールの定義に使用されます。
- DialPlanScript — [回線 1(Line 1)] および [回線 2(Line 2)] のダイヤル プランの指定に使用される構文のスクリプトを作成します。

(注)

- <Par Name> は、設定パラメータの名前を表します。プロファイルでは、**Par_Name** のように、スペースをアンダースコア “_” に置き換えて対応するタグが作成されます。
- 空のデフォルト値フィールドは、空の文字列 < **“”** > を意味します。
- Cisco IP Phone は、特定のプロファイルに存在しないタグの最後の設定値を使用し続けます。
- テンプレートは、指定された順序で比較されます。最初に、*not the closest* で、一致が選択されます。パラメータ名は完全に一致する必要があります。
- プロファイル内のあるパラメータに複数の定義が指定されている場合、Cisco IP Phone ではファイル内の最後の定義が有効になります。
- 空のパラメータ値でパラメータを指定すると、指定されたパラメータは強制的にデフォルト値に戻されます。代わりに空の文字列を指定するには、パラメータ値として空の文字列 **""** を使用します。

プロファイル更新とファームウェア アップグレード

Cisco IP Phone は、セキュアなリモート プロビジョニング（設定）とファームウェア アップグレードをサポートします。プロビジョニングされていない電話は、そのデバイスを対象とする暗号化プロファイルを受信できます。SSL 機能を使用するセキュアな最初のプロビジョニング機能があるため、電話は明示的なキーを必要としません。

プロファイル更新やファームウェア アップグレードを開始または完了するのに、ユーザによる介入は必要ありません。以前のリリースから将来のアップグレード状態に到達するため、中間アップグレードが必要になった場合、Cisco IP Phone のアップグレード ロジックは、マルチステージ アップグレードを自動化できます。プロファイルの再同期は、再同期がソフトウェアのリブートをトリガーして通話が切断される可能性があるため、Cisco IP Phone がアイドル状態のときにのみ試行されます。

汎用パラメータは、プロビジョニング プロセスを管理します。各 Cisco IP Phone は、通常のプロビジョニング サーバ（NPS）と定期的にコンタクトをとるよう設定することができます。NPS と通信する場合、共有秘密キーを使用して更新されたプロファイルを暗号化するため、セキュア プロトコルを使用する必要はありません。NPS には、クライアント証明書を備えた標準的な TFTP、HTTP または HTTPS サーバのいずれかを使用できます。

管理者は、電話の Web ユーザ インターフェイス を使用して、Cisco IP Phone をアップグレード、リブート、再起動、または再起動することができます。また管理者は、これらのタスクを、SIP Notify メッセージを使用して実行することもできます。

設定プロファイルは、サービス プロバイダーのプロビジョニング システムと統合される、一般的なオープン ソース ツールを使用して生成されます。

関連項目

- プロファイル更新の許可と設定（2-21 ページ）
- ファームウェア アップグレードの許可と設定（2-22 ページ）

プロファイル更新の許可と設定

プロファイル更新は、指定された間隔で許可できます。更新されたプロファイルは、TFTP、HTTP、または HTTPS を使用してサーバから電話に送信されます。

ステップ 1　[管理者ログイン（Admin Login）] > [詳細（advanced）] > [音声（Voice）] > [プロビジョニング（Provisioning）] とクリックします。

ステップ 2　[設定プロファイル（Configuration Profile）] セクションで、[プロビジョン有効（Provision Enable）] ドロップダウン リスト ボックスから [はい（Yes）] を選択します。

ステップ 3　パラメータを入力します。

ステップ 4　[すべての変更を送信（Submit All Changes）] をクリックします。

ファームウェア アップグレードの許可と設定

ファームウェアのアップデートは、指定された間隔で許可できます。更新されたファームウェアは、TFTP または HTTP を使用してサーバから電話に送信されます。ファームウェアには個人情報が含まれていないため、ファームウェア アップグレードでセキュリティは問題になりません。

ステップ 1　[管理者ログイン（Admin Login）] > [詳細（advanced）] > [音声（Voice）] > [プロビジョニング（Provisioning）] とクリックします。

ステップ 2　[ファームウェア アップグレード（Firmware Upgrade）] セクションで、[アップグレード有効化（Upgrade Enable）] ドロップダウン リスト ボックスから [はい（Yes）] を選択します。

ステップ 3　パラメータを入力します。

ステップ 4　[すべての変更を送信（Submit All Changes）] をクリックします。

tftp/http/https によるファームウェア アップグレード

3PCC は、tftp/http/https による単一のイメージ アップグレードをサポートします。

（注）　デバイス（新しいベースおよび DCU を含む）は、9.3(3)など、以前のファームウェア リリースにはダウングレードしない可能性があります。詳細については、最新のリリース ノート ドキュメント『*Cisco IP Phone 7800 Series Multiplatform Phones Release Notes*』または『*Cisco IP Phone 8800 Series Multiplatform Phones Release Notes*』のいずれかで、ハードウェア情報およびファームウェアとハードウェアの互換性に関する情報を参照してください。

はじめる前に

ファームウェア ロード ファイルは、アクセス可能なサーバにダウンロードする必要があります。

ステップ 1　3PCC イメージの名前を次のように変更します。

cp-x8xx-sip.aa-b-c-3PCC.cop
変更後
cp-x8xx-sip.aa-b-c-3PCC.tar.gz

引数の説明

X8xx は電話シリーズ（8800 や 7800 など）です。

aa-b-c はリリース番号（10-4-1 など）です。

ステップ 2　“tar –xzvf” コマンドを使用して tar ボールを解凍します。

ステップ 3　フォルダを tftp/http/https のダウンロード ディレクトリにコピーします。

ステップ 4　電話の Web ユーザ インターフェイス で、[管理者ログイン（Admin Login）] > [詳細（advanced）] > [音声（Voice）] > [プロビジョニング（Provisioning）] と移動します。

ステップ 5　ロード ファイル名を検索し、有効な URL に追加します。

ステップ 6　[すべての変更を送信（Submit All Changes）] をクリックします。

ブラウザ コマンドによるファームウェア アップグレード

ブラウザのアドレス バーに入力されたアップグレード コマンドを使用して、電話のファームウェアをアップグレードすることができます。電話は、アイドル状態の場合にのみ更新されます。更新は、コールが完了すると自動的に試行されます。

ステップ 1　Web ブラウザの URL で Cisco IP Phone CP-78xx-3PCC をアップグレードするには、次のコマンドを入力します。

```
http://<phone_ip>/admin/upgrade?<schema>://<serv_ip[:port]>/filepath
```