



设置格式

- [配置配置文件](#)，第 1 页
- [配置文件格式](#)，第 1 页
- [开放配置文件 \(XML\) 的压缩和加密](#)，第 5 页
- [将配置文件应用到电话](#)，第 11 页
- [配置参数类型](#)，第 12 页
- [数据类型](#)，第 18 页
- [配置文件更新和固件升级](#)，第 21 页

配置配置文件

电话接受 XML 格式的配置。

本文档中的示例使用采用 XML 格式 (XML) 语法的配置文件。

有关您的电话的详细信息，请参阅对应的管理指南。每个指南介绍了可通过管理 web 服务器配置参数。

配置文件格式

配置文件定义电话的参数值。

配置文件 XML 格式使用标准 XML 编写工具编译参数和值。



注释 仅支持 UTF-8 字符集。如果您在编辑器中修改配置文件，不要更改编码格式；否则电话将无法识别文件。

每部电话有不同的功能组合，因此，参数组合也不尽相同。

XML 格式 (XML) 配置文件

开放格式配置文件是元素层次结构中具有类似 XML 语法的文本文件，具有元素属性和值。通过这种格式，您可以使用标准工具来创建配置文件。重新同步操作期间，这种格式的配置文件可以从设置服务器发送到电话。文件可以在不编译为二进制对象的情况下发送。

电话可以接受标准工具生成的配置格式。有了此功能后，便不需要再开发从现有数据库生成配置文件的后端设置服务器软件。

为保护配置文件中的机密信息，设置服务器会将这种类型的文件通过 TLS 保护的通道发送至电话。（可选）可以使用 `gzip deflate` 算法 (RFC1951) 压缩文件。

可以使用以下加密方法之一加密文件：

- AES-256-CBC 加密
- 使用 AES-128-GCM 加密算法的基于 RFC-8188 之 HTTP 内容加密

示例：开放配置文件格式

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

`<flat-profile>` 元素标签包含电话可识别的所有参数元素。

配置文件组成要素

配置文件可以包含以下要素：

- 元素标签
- 属性
- 参数
- 格式特性
- XML 备注

元素标签属性

- XML 设置格式和 Web UI 允许配置相同的设置。XML 标签名称和 Web UI 中的字段名类似，但由于存在 XML 元素名称限制，并不尽相同。例如，使用下划线 (`_`) 而不是 `"`。
- 电话会识别具有封装在特殊 `<flat-profile>` 元素中的适当参数名称的元素。
- 元素名称包含在尖括号中。

- 大多数元素名称与设备管理网页中的字段名称类似，进行了以下修改：
 - 元素名称可能不包含空格或特殊字符。要从管理 Web 字段名称获得元素名称，用下划线替换每个空格或特殊字符 [、]、(、) 或 /。
 - 示例：元素 <Resync_On_Reset> 代表重置时重新同步字段。
 - 每个元素名称必须唯一。在管理网页，相同字段可能出现在多个网页上，例如“线路”、“用户”和“分机”页面。将 [n] 附加至元素名称，以指示页面选项卡中显示的号码。
 - 示例：<Dial_Plan_1_> 元素表示第 1 行的拨号计划。
- 每个开始元素标签必须具有匹配的结束元素标签。例如：

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

- 元素标签区分大小写。
- 允许使用空白元素标签，会被解读为将值配置为空。输入不含对应元素标签的开始元素标签，然后在右尖括号 (>) 之前插入空格和斜线。在此示例中，配置文件规则 B 为空：

```
<Profile_Rule_B />
```

- 可以使用空的元素标签来防止重新同步期间改写用户提供的任何值。在以下示例中，用户快捷拨号设置不变：

```
<flat-profile>
<Speed_Dial_2_Name ua="rw"/>
<Speed_Dial_2_Number ua="rw"/>
<Speed_Dial_3_Name ua="rw"/>
<Speed_Dial_3_Number ua="rw"/>
<Speed_Dial_4_Name ua="rw"/>
<Speed_Dial_4_Number ua="rw"/>
<Speed_Dial_5_Name ua="rw"/>
<Speed_Dial_5_Number ua="rw"/>
<Speed_Dial_6_Name ua="rw"/>
<Speed_Dial_6_Number ua="rw"/>
<Speed_Dial_7_Name ua="rw"/>
<Speed_Dial_7_Number ua="rw"/>
<Speed_Dial_8_Name ua="rw"/>
<Speed_Dial_8_Number ua="rw"/>
<Speed_Dial_9_Name ua="rw"/>
<Speed_Dial_9_Number ua="rw"/>
</flat-profile>
```

- 使用空值将相应参数设置为空字符串。它们之间输入不包含任何值的开始和结束元素。在以下示例中，GPP_A 参数设置为空字符串。

```
<flat-profile>
<GPP_A>
```

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

- 无法识别的元素名称将被忽略。

参数属性

这些属性适用于参数：

- 任何并非由配置文件指定的参数在电话中保持不变。
- 无法识别的参数将被忽略。
- 如果开放格式配置文件包含多次出现的相同参数标签，则最后一次出现的标签将覆盖较早出现的标签。为避免无意中改写参数的配置值，我们建议每个配置文件至多指定参数的一个实例。
- 最后处理的配置文件优先。如果多个配置文件指定相同的配置参数，则靠后的配置文件的值优先。

字符串格式

这些属性适用于字符串的格式设置：

- 允许通过标准 XML 语法评论。

```
<!-- My comment is typed here -->
```
- 为便于阅读，允许添加前导和尾随空格，但会从参数值删除。
- 值中的新行字符将转换为空格。
- 允许 <? ?>形式的 XML 标头，但电话会将其忽略。
- 要输入特殊字符，可使用下表所示的基本 XML 字符转义码。

| 特殊字符 | XML 转义码序列 |
|----------|-----------|
| & (& 符号) | & |
| < (小于) | < |
| > (大于) | > |
| ' (撇号) | ' |
| " (双引号) | " |

在以下示例中，输入的字符转义码代表拨号方案规则中必填的大于和小于符号。此示例中定义了信息热线拨号方案，将参数 <Dial_Plan_1_> (管理员登录 > 高级 > 语音 > 分机 (n)) 设置为等于 (S0 <:18005551212>)。

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

```
</Dial_Plan_1_>
</flat-profile>
```

- 使用十进制和十六进制值 (s.a. `(` 和 `.`) 的数字字符转义码被翻译。
- 电话固件仅支持 ASCII 字符。

开放配置文件 (XML) 的压缩和加密

可以压缩开放配置文件以减少设置服务器上的网络负载。也可以对配置文件加密以保护机密信息。压缩并非必需，但其必须在加密之前完成。

开放配置文件压缩

支持的压缩方法是 `gzip deflate` 算法 (RFC1951)。Internet 站点上提供有 `gzip` 实用程序以及实施相同算法 (`zlib`) 的压缩库。

为识别压缩，电话期望压缩文件中包含 `gzip` 兼容标头。调用原始开放配置文件上的 `gzip` 实用程序会生成标头。电话会检查所下载文件的标头以确定文件格式。

例如，如果 `profile.xml` 是有效的配置文件，则文件 `profile.xml.gz` 亦可接受。以下任一命令可生成此配置文件类型：

- `>gzip profile.xml`
原始文件替换为压缩文件。
- `>cat profile.xml | gzip > profile.xml.gz`
保留原始文件不变，生成新的压缩的文件。

[通过 Gzip 压缩开放配置文件](#)部分提供了压缩教程。

开放配置文件加密

无论文件是否压缩，都可以使用对称密钥加密来对开放配置文件加密。但如果应用压缩，则压缩必须在加密前应用。

部署后，设置服务器使用 `HTTPS` 来处理电话的初次设置。预加密配置文件离线可允许随后使用 `HTTP` 来重新同步配置文件。如此可在大型部署中减少 `HTTPS` 服务器上的负载。

电话支持两种配置文件加密方法：

- AES-256-CBC 加密
- 使用 AES-128-GCM 加密算法的基于 RFC 8188 之 `HTTP` 内容加密

密钥或输入密钥材料 (IKM) 必须及早预部署到设备。通过使用 `HTTPS` 可以安全地完成密钥的引导。配置文件名不需要特定的格式，但以 `.cfg` 扩展名结尾的文件名通常表示配置文件。

AES-256-CBC 加密

电话支持配置文件的 AES-256-CBC 加密。

OpenSSL 加密工具（可从多个 Internet 站点下载）可以执行加密。支持 256 位 AES 加密可能需要重新编译工具来启用 AES 代码。已根据版本 openssl-1.1.1d 测试固件。

使用 [OpenSSL 加密配置文件](#) 提供加密教程。

对于加密文件，配置文件期望文件的格式与通过以下命令生成的文件相同：

```
# example encryption key = SecretPhrase1234

openssl enc -aes-256-cbc -md md5 -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
```

密钥前加有小写字母 **-k**，可以是任何纯文本短语，用于生成随机的 64 位盐。利用 **-k** 参数指定的密钥，加密工具可得出随机的 128 位初始化向量和实际的 256 位加密密钥。

如果对配置文件使用这种形式的加密，电话必须了解密钥值，以对文件进行解密。此值在配置文件 URL 中被指定为限定符。语法使用显式 URL，如下所示：

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

使用一个 **Profile_Rule** 参数设置此值。

宏扩展

评估之前多个设置参数会进行内部宏扩展。此预评估为控制电话重新同步和升级活动提供了更高的灵活性。

评估之前这些参数组会进行宏扩展：

- Resync_Trigger_*
- Profile_Rule*
- Log_xxx_Msg
- Upgrade_Rule

在特定情况下，某些通用参数 (GPP_*) 也会进行宏扩展，如 [可选重新同步参数](#)，第 10 页中明确所示。

宏扩展期间，指定变量的内容会替换 \$NAME 和 \$(NAME) 形式的表达式。这些变量包括通用参数、多个产品标识符、特定事件计时器和设置状态值。如需完整列表，请参阅 [宏扩展变量](#)。

在以下示例中，表达式 \$(MAU) 用于插入 MAC 地址 000E08012345。

管理员输入：\$ (MAU) config.cfg

MAC 地址为 000E08012345 的设备的宏扩展结果是：000E08012345config.cfg

如果无法识别宏名称，其将保持不扩展。例如，名称 `STRANGE` 不会被识别为有效的宏名称，`MAU` 则会被识别为有效的宏名称。

管理员输入：`$STRANGE$MAU.cfg`

MAC 地址为 `000E08012345` 的设备的宏扩展结果是：`$STRANGE000E08012345.cfg`

宏扩展不可递归应用。例如，`$$MAU` 将扩展为 `$MAU`（`$$` 被扩展），不会生成 MAC 地址。

特殊用途参数 `GPP_SA` 至 `GPP_SD` 的内容将映射到宏表达式 `$$SA` 至 `$$SD`。这些参数将仅宏扩展为重新同步 URL 中 `--key`、`--uid` 和 `--pwd` 选项的参数。

条件表达式

条件表达式可以触发重新同步事件，并从备用 URL 中挑选一个用于重新同步和升级操作。

条件表达式包含以 `and` 运算符分隔的比较列表。所有比较必须满足条件为 `true`。

每个比较可以涉及以下三种文字类型之一：

- 整数值
- 软件或硬件版本号
- 双引号字符串

版本号

具有多平台固件的 Cisco IP 电话的软件版本使用以下格式（其中 *BN* 为内部版本号）：

- 对于固件版本 `11.3(1)SR1` 及其之前的版本：`sipyyyy.11-0-1MPP-376`

其中 `yyyy` 表示电话型号或电话系列；`11` 是主要版本；`0` 是次要版本；`1MPP` 是微版本；`376` 是内部版本号。

- 对于固件版本 `11.3(2)` 及其之后的版本：`sipyyyy.11-3-2MPP0001-609`

其中 `yyyy` 表示电话型号或电话系列；`11` 是主要版本；`3` 是次要版本；`2MPP0001` 是微版本；`609` 是内部版本号。

比较字符串必须使用相同的格式。否则，会导致格式解析错误。

比较软件版本时，按顺序比较主要版本、次要版本和微版本，最左边的数字优先于后面的数字。当版本号相同时，将比较内部版本号。

有效版本号的示例

- 对于固件版本 `11.3(1)SR1` 及其之前的版本：

```
sip7832.11-0-1MPP-312
```

- 对于固件版本 `11.3(2)` 及其之后的版本：

```
sip7832.11-3-2MPP0001-609
```

- 对于固件版本 `11.3(1)SR1` 及其之前的版本：

```
sip8832.11-0-1MPP-312
```

- 对于固件版本 11.3(2) 及其之后的版本:

```
sip8832.11-3-2MPP0001-609
```

比较

- 对于固件版本 11.3(1)SR1 及其之前的版本:

```
sipyyyy.11-3-1MPP-110      sipyyyy.11-2-3MPP-256
```

- 对于固件版本 11.3(2) 及其之后的版本:

```
sipyyyy.11-3-2MPP0002-256      sipyyyy.11-3-2MPP0001-609
```

可以比较引用的字符串是相等还是不等。也可以在算术上比较整数和版本号。比较运算符可以用符号或首字母缩写表示。首字母缩写可方便地在开放格式配置文件中表达条件。

| 运算符 | 备用语法 | 说明 | 适用于整数和版本运算对象 | 适用于加引号对象 |
|-----|------|-------|--------------|----------|
| = | eq | 等于 | 是 | 是 |
| != | ne | 不等于 | 是 | 是 |
| < | lt | 小于 | 是 | 否 |
| <= | le | 小于或等于 | 是 | 否 |
| > | gt | 大于 | 是 | 否 |
| >= | ge | 大于或等于 | 是 | 否 |
| AND | | 和 | 是 | 是 |

在需要字符串文字的地方，必须将宏变量置于双引号中，这一点至关重要。不要在需要数字或版本号的地方这样做。

如果是用于 Profile_Rule* 和 Upgrade_Rule 参数，必须如此升级规则示例所述，将条件表达式置于语法“(expr)?”内。请记住，将 BN 替换为要升级到的固件负载内部版本号。

- 对于固件版本 11.3(1)SR1 及其之前的版本

```
($SWVER ne sip7832.11-0-0MPP-256)? http://ps.tell.com/sw/sip7832.11-0-0MPP-BN.loads
```

- 对于固件版本 11.3(2) 及其之后的版本

```
($SWVER ne sip7832.11-3-2MPP0001-609)?  
http://ps.tell.com/sw/sip7832.11-3-2MPP0001-BN.loads
```

- 对于固件版本 11.3(1)SR1 及其之前的版本

```
($SWVER ne sip8832.11-0-0MPP-256)? http://ps.tell.com/sw/sip8832.11-0-0MPP-BN.loads
```


- 对于固件版本 11.3(2) 及其之后的版本

```
($SWVER ne sip8832.11-3-2MPP0001-609)?
http://ps.tell.com/sw/sip8832xx.11-3-2MPP0001-BN.loads
```

配置 `Resync_Trigger_*` 参数时，不要将前置语法放在括号中。

URL 语法

使用标准 URL 语法可指定如何分别在 `Profile_Rule*` 和 `Upgrade_Rule` 参数中检索配置文件和固件负载。语法如下：

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

其中，`scheme` 是下列值之一：

- tftp
- http
- https

如果 `scheme` 被忽略，则假设为 tftp。服务器可以是 DNS 识别的主机名或数字 IP 地址。端口是目标 UDP 或 TCP 端口号。文件路径必须以根目录 (/) 开头；其必须是绝对路径。

如果 `server` 缺失，将通过 DHCP（选项 66）指定 tftp 服务器。



注释 有关升级规则，必须指定服务器。

如果 `port` 缺失，将使用指定方案的标准端口。Tftp 使用 UDP 端口 69，http 使用 TCP 端口 80，https 使用 TCP 端口 443。

必须存在文件路径。它不必引用静态文件，但可以指明通过 CGI 获取的动态内容。

URL 内将应用宏扩展。以下为有效 URL 的示例：

```
/$MA.cfg
/cisco/cfg.xml
192.168.1.130/profiles/init.cfg
tftp://prov.call.com/cpe/cisco$MA.cfg
http://neptune.speak.net:8080/prov/$D/$E.cfg
https://secure.me.com/profile?Linksys
```

使用 DHCP 选项 66 时，空白语法不受升级规则支持。它仅适用于 `Profile Rule*`。

基于 RFC 8188 的 HTTP 内容加密

电话支持使用 AES-128-GCM 加密算法对配置文件进行基于 RFC 8188 的 HTTP 内容加密。使用此加密方法，任何实体均可读取 HTTP 消息标头。但是，只有知道输入密钥材料 (IKM) 的实体能够读取有效负载内容。使用 IKM 部署电话时，电话与部署服务器可以安全地交换配置文件，同时允许第三方网络元素出于分析和监控目的使用消息标头。

XML 配置参数 **IKM_HTTP_Encrypt_Content** 可在电话上保留 IKM。出于安全原因，此参数在电话管理网页上无法访问。它也不会电话的配置文件中显示，后者可从电话的 IP 地址或从发送到部署服务器的电话配置报告进行访问。

如果想要使用基于 RFC 8188 的加密，确保满足以下条件：

- 通过使用从部署服务器发送到电话的配置文件中的 XML 参数 **IKM_HTTP_Encrypt_Content** 指定 IKM，利用 IKM 部署电话。
- 如果此加密应用于从部署服务器发送到电话的配置文件，确保配置文件中的 *Content-Encoding* HTTP 标头中包含 “aes128gcm”。

如果没有此标头，则 AES-256-CBC 方法优先。如果配置文件规则中存在 AES-256-CBC 密钥，则电话应用 AES-256-CBC 解密，无论 IKM 如何。

- 如果想要电话将此加密应用到发送给部署服务器的配置报告，请确保报告规则中没有指定 AES-256-CBC 密钥。

可选重新同步参数

可选参数 **key**、**uid** 和 **pwd** 可置于 **Profile_Rule*** 参数中输入的 URL 之前，一起放在方括号内。

key

本 **--key** 选项告知电话从部署服务器接收的配置文件使用 AES-256-CBC 加密进行加密，除非文件中的 *Content-Encoding* 标头指示 “aes128gcm” 加密。密钥本身被指定为后跟术语 **--key** 的字符串。可选择将密钥括在双引号 (") 内。电话使用密钥解密配置文件。

使用示例

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

括号内的可选参数是宏扩展的。仅当用作密钥选项参数时，特殊用途参数 **GPP_SA** 至 **GPP_SD** 才会宏扩展为宏变量 **\$SA** 至 **\$SD**。请参阅以下示例：

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

在开放格式配置文件中，**--key** 的参数必须与指定给 **openssl** 的 **-k** 选项的参数相同。

uid 和 pwd

uid 和 **pwd** 选项可用于指定当请求指定的 URL 时，为响应 HTTP 基本和摘要式身份验证质询而发送的用户 ID 和密码。括号内的可选参数是宏扩展的。仅当用作密钥选项参数时，特殊用途参数 **GPP_SA** 至 **GPP_SD** 才会宏扩展为宏变量 **\$SA** 至 **\$SD**。请参阅以下示例：

```
GPP_SA = MyUserID  
GPP_SB = MySecretPassword
```

```
[--uid $SA --pwd $SB] https://provisioning_server_url/path_to_your_config/your_config.xml
```

随后将扩展为：

```
[--uid MyUserID --pwd MySecretPassword]  
https://provisioning_server_url/path_to_your_config/your_config.xml
```

将配置文件应用到电话

创建 XML 配置脚本后，必须将其传送给电话以供应用。要应用配置，您可以使用 Web 浏览器或 cURL 命令行实用程序，从 TFTP、HTTP 或 HTTPS 服务器将配置文件下载到电话。

从 TFTP 服务器将配置文件下载到电话

完成这些步骤以将配置文件下载到 PC 上的 TFTP 服务器应用程序。

过程

步骤 1 将您的 PC 连接到电话 LAN。

步骤 2 在 PC 上运行 TFTP 服务器应用程序，并确保配置文件在 TFTP 根目录中可用。

步骤 3 在 Web 浏览器中，输入电话 LAN IP 地址、计算机的 IP 地址、文件名以及登录凭证。使用此格式：

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

示例：

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

通过 cURL 将配置文件下载到电话

遵循以下步骤，使用 cURL 将配置下载到电话。此命令行工具用于通过 URL 语法传输数据。要下载 cURL，请访问：

<https://curl.haxx.se/download.html>



注释 我们建议您不要使用 cURL 将配置发布到电话，因为使用 cURL 时用户名和密码可能会遭截获。

过程

步骤 1 将您的 PC 连接到电话的 LAN 端口。

步骤 2 输入以下 cURL 命令以将配置文件下载到电话：

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

配置参数类型

本节介绍根据功能大致编排的设置参数：

存在以下设置参数类型：

- 一般用途
- 启用
- 触发器
- 可配置的日程表
- 配置文件规则
- 升级规则

通用参数

当将电话配置为与特定设置服务器解决方案进行交互时，通用参数 GPP_*（管理员登录 > 高级 > 语音 > 设置）用作免费字符串寄存器。GPP_* 参数默认为空。可将它们配置为包含不同的值，其中包括：

- 加密密钥
- URL
- 多阶段设置状态信息。
- Post 请求模板
- 参数名称别名映射
- 最终并入完整的参数值的部分字符串值。

GPP_* 参数可在其他设置参数内用于宏扩展。出于此目的，一个字母的大写宏名称（A through P）足以识别 GPP_A 至 GPP_P 的内容。另外，当用作以下 URL 选项的参数时，两个字母的大写宏名称 SA 至 SD 将 GPP_SA 至 GPP_SD 标识为特殊情况：

key、uid 和 pwd

这些参数可在设置和升级规则中用作变量。引用它们时需在变量名称前添加 '\$' 字符，例如 \$GPP_A。

使用通用参数

例如，如果 GPP_A 包含字符串 ABC，GPP_B 包含 123，表达式 \$A\$B 将宏将扩展到 ABC123。

开始之前

访问电话管理网页。请参阅：[访问电话 Web 界面](#)。

过程

- 步骤 1** 选择语音 > 部署。
- 步骤 2** 滚动至 **General Purpose Parameters** 部分。
- 步骤 3** 在 GPP A 到 GPP P 的字段中输入有效值。
- 步骤 4** 单击 **Submit All Changes**。

启用参数

Provision_Enable 和 Upgrade_Enable 参数控制着所有配置文件重新同步和固件升级操作。这些参数彼此独立控制重新同步和升级。这些参数还控制通过管理 web 服务器发出的重新同步和升级 URL 命令。这两个参数默认都设置为是。

Resync_From_SIP 参数控制重新同步操作请求。SIP NOTIFY 事件从服务提供商代理服务器发送到电话。如果启用，代理可以请求重新同步。为此，代理会发送 SIP NOTIFY 消息到设备，其中包含 Event: resync 标头。

设备以 401 响应消息质询请求（授权遭拒，因凭证已被使用）。设备在支持来自代理的重新同步请求之前，预期会出现经过验证的后续请求。Event: reboot_now 和 Event: restart_now 标头会分别执行冷重启和热重启，也会遭到质询。

其余两个会启用 Resync_On_Reset 和 Resync_After_Upgrade_Attempt。这些参数确定在开机软件重新启动后以及每次升级尝试后，设备是否执行重新同步操作。

启用 Resync_On_Reset 后，在执行重置之前，设备在启动序列后引入随机延迟。延迟时间是随机的，上限为 Resync_Random_Delay 指定的值（以秒为单位）。在同时开机的电话池中，这一延迟将传播到每个设备重新同步请求的开始时间。在大型住宅部署中，如果发生地区性电力故障，此功能将十分有用。

触发器

电话允许您按特定时间间隔或在特定时间重新同步。

在特定时间间隔重新同步

电话可定期与设置服务器重新同步。重新同步间隔在 `Resync_Periodic`（秒）中配置。如果此值为空，设备不会定期重新同步。

重新同步通常在语音线路空闲时执行。如果语音线路处于活动状态且需要重新同步，电话将延迟重新同步过程，直到线路再次变为空闲。重新同步可能导致配置参数的值更改。

因为电话不能从服务器检索配置文件、下载的文件损坏或者出现内部错误，重新同步操作可能会失败。`Resync_Error_Retry_Delay`（秒）中指定的时间后，设备将尝试再次重新同步。如果 `Resync_Error_Retry_Delay` 设置为 0，在尝试重新同步失败后，设备不会再尝试重新同步。

如果升级失败，会在 `Upgrade_Error_Retry_Delay` 中指定的秒数后重试。

有两个可配置参数可用于有条件触发重新同步：`Resync_Trigger_1` 和 `Resync_Trigger_2`。每个参数都可以使用经过宏扩展的条件表达式设置。重新同步时间间隔过期时（下次重新同步的时间），触发器（如有设置）将阻止重新同步，除非一个或多个触发器的计算结果为 `true`。

以下示例条件将触发重新同步。在此示例中，最后一次电话升级尝试已过去超过 5 分钟（300 秒），并且自上次重新同步尝试以来已经过去至少 10 分钟（600 秒）。

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

在特定时间重新同步

`Resync_At` 参数可让电话在特定时间重新同步。此参数使用 24 小时制 (hhmm) 指定时间。

`Resync_At_Random_Delay` 参数允许电话在未指定的延迟时间重新同步。此参数用于正整数格式指定时间。

应避免来自多部电话、设定为在相同时间重新同步的重新同步请求涌向服务器。为此，电话会在指定时间后最多 10 分钟触发重新同步。

例如，如果您将重新同步时间设置为 1000（上午 10:00），电话会在上午 10:00 与 10:10 之间的任何时间触发重新同步。

默认情况下，此功能处于禁用状态。如果设置了 `Resync_At` 参数，`Resync_Periodic` 参数会被忽略。

可配置的日程表

您可以配置日程表以定期执行重新同步，并且可以使用以下设置参数指定重新同步和升级失败时的重试间隔：

- `Resync_Periodic`
- `Resync_Error_Retry_Delay`
- `Upgrade_Error_Retry_Delay`

每个参数接受单一的延迟值（秒）。新的扩展语法允许以逗号分隔的连续延迟元素列表。序列中的最后一个元素会永久隐式重复。

（可选）您可以使用 + 号指定附加随机额外延迟的另一个数值。

示例 1

在此示例中，电话会每 2 小时定期重新同步一次。如果重新同步失败，设备将按这些时间间隔重试：30 分钟、1 小时、2 小时、4 小时。之后，设备会继续每隔 4 小时尝试一次，直至重新同步成功。

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

示例 2

在此示例中，设备每小时（加上最多 10 分钟的额外随机延迟）定期重新同步一次。如果重新同步失败，设备将按以下时间间隔重试：30 分钟（最多再加 5 分钟）、1 小时（最多再加 10 分钟）、2 小时（最多再加 15 分钟）。之后，设备会继续每隔 2 小时（最多再加 15 分钟）尝试一次，直至重新同步成功。

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

示例 3

在此示例中，如果远程升级尝试失败，设备会依次间隔 30 分钟、1 小时、2 小时重试升级。如果升级仍然失败，设备会每隔 4 至 5 小时重试，直至升级成功。

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

配置文件规则

电话提供多个远程配置文件参数 (Profile_Rule*)。因此，每个重新同步操作都可以取回由不同服务器管理的多个文件。

在最简单的情形中，设备会定期重新同步到中央服务器上的单个配置文件，从而更新所有相关内部参数。也可以将配置文件拆分成多个不同的文件。有一个适用于部署中的所有电话的通用文件。还会为每个帐户提供单独、唯一的文件。加密密钥和证书信息仍可由存储在单独服务器上的另一个配置文件提供。

只要重新同步操作到期，电话就会按顺序评估四个 Profile_Rule* 参数：

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

每次评估可能会导致自远程设置服务器检索配置文件，其中可能会更新一定数量的内部参数。如果评估失败，重新同步序列将中断，并按照 Resync_Error_Retry_Delay 参数（秒）指定的值从头再次重

试。如果所有评估都成功，设备将等待 `Resync_Periodic` 参数指定的秒数，然后执行另一个重新同步。

每个 `Profile_Rule*` 参数内容中都包含一组备用值。备用值之间以 |（竖线）字符分隔。每个备用值包含一个条件表达式、一个赋值表达式、一个配置文件 URL 和任何关联的 URL 选项。在每个备用值中，所有这些要素都是可选的。以下是有效的组合，并且必须依此顺序显示（如果有）：

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

在每个 `Profile_Rule*` 参数内，除最后一个备用值外，其他所有备用值都必须提供一个条件表达式。系统会按如下方式评估及处理此表达式：

1. 从左到右评估条件，直到找到评估结果为 `true` 的值（或直至找到一个不含条件表达式的备用值）。
2. 评估任何随同的赋值表达式（如果有）。
3. 如果 URL 被指定为该备用值的一部分，将尝试下载位于指定 URL 的配置文件。系统会尝试相应更新内部参数。

如果所有备用值都有条件表达式且没有条件被评估为 `true`（或者如果整个配置文件规则为空），则会跳过整个 `Profile_Rule*` 参数。接下来会评估序列中的下一个配置文件规则参数。

示例 1

本示例将无条件重新同步到指定 URL 上的配置文件，并执行到远程设置服务器的 HTTP GET 请求：

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

示例 2

在本例中，设备重新同步到两个不同的 URL，具体取决于第 1 行的注册状态。在丢失注册的情况下，设备针对 CGI 脚本执行 HTTP POST。设备将发送宏扩展 `GPP_A` 的内容，可能提供有关设备状态的更多信息：

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

示例 3

在此示例中，设备将重新同步到相同服务器。如果未在设备上安装证书（对于 2.0 版之前的旧设备），设备将提供更多信息：

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


示例 4

在此示例中，在将 GPP_A 设定为通过第一个 URL 设置的值之前，线路 1 会一直禁用。然后，它会重新同步到第二个 URL：

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

示例 5

在此示例中，假设服务器返回的配置文件包含 XML 元素标签。这些标签必须按照 GPP_B 中存储的别名映射图，重新映射到适当的参数名称：

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

如果未从服务器收到请求的配置文件，重新同步通常被视为失败。Resync_Fails_On_FNF 参数可以改写此默认行为。如果 Resync_Fails_On_FNF 设置为“否”，设备收到服务器发回的“未找到文件”响应视为重新同步成功。Resync_Fails_On_FNF 的默认值为“是”。

升级规则

升级规则是告诉设备激活一个新的负载，以及必要时从哪里获得负载。如果负载已在设备上，则其不会尝试获取负载。因此，当所需的负载处于非活动分区时，负载位置的有效性并不重要。

Upgrade_Rule 指定不同于当前负载时将下载和应用的固件负载，除非被条件表达式限制或 Upgrade_Enable 设定为否。

电话提供一个可配置远程升级参数 Upgrade_Rule。此参数接受与配置文件规则参数类似的语法。升级不支持 URL 选项，但可以使用条件表达式和赋值表达式。如果使用条件表达式，此参数可填充为多个以 | 字符分隔的备用值。每个备用值的语法如下：

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

如果采用 Profile_Rule* 参数，Upgrade_Rule 参数将评估每个备用值，直到满足条件表达式或备用值无条件表达式。评估随同的赋值表达式（如有指定）。然后，尝试升级到指定的 URL。

如果 Upgrade_Rule 包含无条件表达式的 URL，设备将升级到 URL 指定的固件映像。宏扩展和规则评估后，设备不会重新尝试升级，直到规则遭修改或方案 + 服务器 + 端口 + 文件路径的有效组合被更改。

要尝试进行固件升级，设备将在程序开始时禁用音频，并在程序结束时重新启动。仅当所有语音线路当前都处于非活动状态时，设备才会自动开始由 Upgrade_Rule 的内容驱动的升级。

例如，

```
https://10.73.10.223/firmware/sip7832.11-3-1MPP-678.loads
```

在此示例中，Upgrade_Rule 将固件升级到指定 URL 中存储的映像。

本例将指示设备根据通用参数 `GPP_F` 的内容，下载两幅图像中的一幅。

设备可以强制执行固件版本号的降级限制，这可能是一个有用的定制选项。如果 `Downgrade_Rev_Limit` 参数中配置了有效的固件版本号，设备将拒绝为低于指定限制的固件版本尝试升级。

数据类型

这些数据类型与配置文件参数一起使用：

- `{a,b,c,...}` — 在 a、b、c 之间选择 ...
- `Bool` — 布尔值为“是”或“否”。
- `CadScript` — 指定信号节律参数的简短说明。最多 127 个字符。

语法：`S1[:S2]`，其中：

- $S_i = D_i(\text{on}_{i,1}/\text{off}_{i,1}[\text{on}_{i,2}/\text{off}_{i,2}[\text{on}_{i,3}/\text{off}_{i,3}[\text{on}_{i,4}/\text{off}_{i,4}[\text{on}_{i,5}/\text{off}_{i,5}[\text{on}_{i,6}/\text{off}_{i,6}]]]])$ ，称为区段。
- $\text{on}_{i,j}$ 和 $\text{off}_{i,j}$ 是片段开/关的持续时间，以秒为单位。i = 1 或 2，j = 1 或 6。
- D_i 是区段的总持续时间，以秒为单位。

所有持续时间最多可有三位小数，提供毫秒级分辨率。通配符“*”表示持续时间无限。区段中的片段将按顺序重复执行，直至执行时间达到总持续时间。

示例 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
```

示例 2 — 独特振铃（短、短、短、长）：

```
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
```

- `DialPlanScript` — 用于指定线路 1 和线路 2 的拨号方案的脚本语法。
- `Float<n>` — 最多小数点后 n 位的浮点值。

- **FQDN** — 完全限定域名。它最多可包含 63 个字符。示例如下：
 - sip.Cisco.com:5060 或 109.12.14.12:12345
 - sip.Cisco.com 或 109.12.14.12
- **FreqScript** — 指定提示音频率和等级参数的简短说明。最多可包含 127 个字符。
语法: $F_1@L_1[F_2@L_2[F_3@L_3[F_4@L_4[F_5@L_5[F_6@L_6]]]]]$, 其中:
 - $F_1 - F_6$ 是频率, 以赫兹为单位 (仅无符号整数)。
 - $L_1 - L_6$ 是相应的等级, 以 dBm 为单位 (最多一位小数)。

逗号之前和之后允许使用空格, 但不建议这样做。

示例 1 — 呼叫等待提示音:

```
440@-10

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

示例 2 — 拨号音:

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

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

- **IP** — 有效的 IPv4 地址, 格式为 x.x.x.x, 其中 x 介于 0 与 255 之间。示例: 10.1.2.100。
- **UserID** — URL 中出现的用户 ID; 最多 63 个字符。
- **Phone** — 电话号码字符串, 例如 14081234567、*69、*72、345678; 或者一般 URL, 例如 1234@10.10.10.100:5068 或 jsmith@Cisco.com。字符串最多可以包含 39 个字符。
- **PhTmpl** — 电话号码模板。每个模板可包含一个或多个用逗号 (,) 分隔的模式。每个模式开头处的空格将被忽略。“?” 和 “*” 表示通配符字符。要从字面上表示, 请使用 %xx。例如, %2a 表示 *。模板最多可以包含 39 个字符。示例: “1408*, 1510*”, “1408123????, 555?1。”
- **Port** — TCP/UDP 端口号 (0-65535)。它可以用十进制或十六进制格式指定。
- **ProvisioningRuleSyntax** — 用于定义配置重新同步和固件升级规则的脚本语法。
- **PwrLevel** — 以 dBm 表示的功率级别, 含一位小数, 例如 -13.5 或 1.5 (dBm)。
- **RscTmpl** — SIP 响应状态代码的模板, 如 “404, 5*”, “61?”, “407, 408, 487, 481”。最多可包含 39 个字符。
- **Sig<n>** — 有符号的 n 位值。它可以用十进制或十六进制格式指定。负值前面必须添加 - 号。正值前面可以选择性添加 + 号。
- **Star Codes** — 补充服务的激活码, 如 *69。代码最多可以包含 7 个字符。

- **Str<n>** — 最多具有 n 个非保留字符的通用字符串。
- **Time<n>** — 持续时间，以秒为单位，最多 n 位小数。额外指定的小数位将被忽略。
- **ToneScript** — 指定电话进展音的频率、电平和节奏参数的简短说明。脚本最多可包含 127 个字符。

语法: **FreqScript**; Z_1 ; Z_2]

在 **CadScript** 中，区段 Z_1 与区段 S_1 类似，除每个开/关片段后跟频率要素参数外: $Z_1 = D_1(\text{on}_{i,1}/\text{off}_{i,1}/f_{i,1}[\text{on}_{i,2}/\text{off}_{i,2}/f_{i,2}[\text{on}_{i,3}/\text{off}_{i,3}/f_{i,3}[\text{on}_{i,4}/\text{off}_{i,4}/f_{i,4}[\text{on}_{i,5}/\text{off}_{i,5}/f_{i,5}[\text{on}_{i,6}/\text{off}_{i,6}/f_{i,6}]]]]]])$ ，其中:

- $f_{i,j} = n_1[+n_2]+n_3[+n_4[+n_5[+n_6]]]]]$ 。
- $1 < n_k < 6$ 指定 **FreqScript** 中用于该片段的频率要素。

如果一个片段中使用了多个频率要素，会将所有要素汇总。

示例 1 — 拨号音:

```
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
```

示例 2 — 通信音:

```
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
```

- **Uns<n>** — 无符号 n 位值，其中 $n = 8$ 、16 或 32。它可以以十进制格式或十六进制格式（例如 12 或 0x18）指定，只要该值可以容纳 n 位。



注释 考虑以下因素：

- <Par Name> 表示配置参数名称。在配置文件中，用下划线 “_” 替换空格即形成相应标签（如 **Par_Name**）。
- 空的缺省值字段表示空字符串 < “ ” >。
- 电话将继续为给定配置文件中未出现的标签使用最后一个配置的值。
- 模板按照给定的顺序进行比较。首先是 *not the closest*，选择匹配项。参数名称必须完全匹配。
- 如果配置文件中为一个参数给出了多个定义，则电话会采用文件中的最后一个定义。
- 具有空参数值的参数规范强制参数返回其默认值。要改为指定空字符串，使用空字符串 "" 作为参数值。

配置文件更新和固件升级

电话支持安全远程设置（配置）和固件升级。未设置的电话可以接收发往该设备的加密配置文件。由于使用 SSL 功能的安全首次设置机制，电话不需要显式密钥。

无需用户介入即可启动或结束配置文件更新或者固件升级，或者必须进行中间升级才能从较旧版本达到未来的升级状态。仅当电话处于空闲状态时才会执行配置文件重新同步，因为重新同步可能触发软件重新启动以及呼叫断开连接。

通用参数管理设置过程。可配置每部电话定期联系一般设置服务器 (NPS)。与 NPS 的通信并不要求一定使用安全协议，因为更新的配置文件经共享密钥加密。NPS 可以是含客户端证书的标准 TFTP、HTTP 或 HTTPS 服务器。

管理员可以使用电话的 Web 用户界面升级、重新引导、重新启动或重新同步电话。管理员还可以使用 SIP notify 消息执行这些任务。

配置文件通过使用与服务提供商设置系统集成的通用开源工具生成。

允许配置文件更新

配置文件更新可以按指定的时间间隔进行。更新的配置文件将通过 TFTP、HTTP 或 HTTPS 从服务器发送到电话。

您还可以使用 XML(cfg.xml) 代码配置电话配置文件中的参数。

开始之前

访问电话管理网页。请参阅：[访问电话 Web 界面](#)。

过程

步骤 1 选择语音 > 部署。

步骤 2 在配置文件部分，从启用设置参数中选择是。

您可以通过输入以下格式的字符串，在电话配置 XML 文件 (cfg.xml) 中配置此参数：

```
<Provision_Enable ua="na">Yes</Provision_Enable>
```

默认值：Yes

步骤 3 如[配置文件重新同步参数](#)表中所述设置参数。

步骤 4 单击 **Submit All Changes**。

允许和配置固件升级

固件更新可以按指定的时间间隔进行。更新的固件将通过 TFTP 或 HTTP 从服务器发送到电话。固件升级中安全性不是一个问题，因为固件不含个人信息。

您还可以使用 XML(cfg.xml) 代码配置电话配置文件中的参数。

开始之前

访问电话管理网页。请参阅：[访问电话 Web 界面](#)。

过程

步骤 1 选择语音 > 部署。

步骤 2 在固件升级部分，从启用升级参数中选择是。

您可以通过输入以下格式的字符串，在电话配置 XML 文件 (cfg.xml) 中配置此参数：

```
<Upgrade_Enable ua="na">Yes</Upgrade_Enable>
```

选项：Yes 和 No

默认值：Yes

步骤 3 设置 **Upgrade Error Retry Delay** 参数（秒）。

如果升级失败，应用升级重试间隔（以秒为单位）。设备有一个固件升级错误计时器，会在固件升级尝试失败后激活。计时器使用此参数中的值进行初始化。当此计时器倒计时至零时，会再次尝试进行固件升级。

您可以通过输入以下格式的字符串，在电话配置 XML 文件 (cfg.xml) 中配置此参数：

```
<Upgrade_Error_Retry_Delay ua="na">3600</Upgrade_Error_Retry_Delay>
```

默认值：3600

:

```
<tftp|http|https>://<ip address>/image/<load name>
```

步骤 4 输入定义升级条件及所关联固件 URL 的固件升级脚本，以设置 **Upgrade Rule** 参数。它使用相同的语法作为配置文件规则。输入脚本并使用以下格式输入升级规则：

```
<tftp|http|https>://<ipaddress>/image/<load name>
```

例如：

```
tftp://192.168.1.5/image/sip88xx.11-0-0MPP-BN.loads
```

```
tftp://192.168.1.5/image/sip78xx.11-0-1MPP-BN.loads
```

您可以通过输入以下格式的字符串，在电话配置 XML 文件 (cfg.xml) 中配置此参数：

```
<Upgrade_Rule ua="na">http://10.74.10.205:6970/sip8845_65.0104-MPP-9875dev.loads  
</Upgrade_Rule>
```

步骤 5 单击 **Submit All Changes**。

通过 TFTP、HTTP 或 HTTPS 升级固件

电话通过 TFTP、HTTP 或 HTTPS 支持固件升级。



注释 降级到早期版本可能并非对所有设备都适用。有关详细信息，请参阅您电话和固件版本的发行说明。

开始之前

必须将固件负载文件下载到可访问的服务器。

过程

步骤 1 将文件夹复制到 TFTP、HTTP 或 HTTPS 下载目录

步骤 2 访问电话管理网页。请参阅：[访问电话 Web 界面](#)。

步骤 3 选择语音 > 部署。

步骤 4 单击 **Submit All Changes**。

使用浏览器命令升级固件

在浏览器地址栏中输入的升级命令可用于在电话上升级固件。电话只会在空闲时更新。呼叫完成后，会自动尝试更新。

过程

要在 Web 浏览器中使用 URL 升级电话，请输入以下命令：

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

当地语言翻译版本说明

思科可能会在某些地方提供本内容的当地语言翻译版本。请注意，翻译版本仅供参考，如有任何不一致之处，以本内容的英文版本为准。