

寄存器在RMS 4.1的新建的IMEI单元使用Python自动脚本通用小信元(USC)

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[配置](#)

[验证](#)

[故障排除](#)

简介

本文在远程管理管理业务(RMS)中描述进程如何注册新的IMEI使用自动脚本。

[先决条件](#)

[要求](#)

本文档没有任何特定的要求。

[使用的组件](#)

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始(默认)配置。如果您使用的是真实网络,请确保您已经了解所有命令的潜在影响。

背景信息

在运行此进程前请检查与客户权限、用户和通行证到远程管理管理业务(RMS)有弹性服务器(root用户)。

在步骤的开头部分和在末端,请检查id池使用情况报告发现多少上升限度留下。

如果没有在池的足够的cellid的重复的cellid在网络可以创建。

配置

运行寄存器步骤：

步骤1.如镜像所显示，根据此格式创建CSV文件，：

A	B	C	D	E
EID	Select Area Manually	Area	RFProfile_New	Activated
001B67-352639055652167	TRUE	Israel	13dBm_Multicell_Open	TRUE

步骤2. FTP CSV文件到RMS有用户/通行证的弹性1服务器

位置库：/intucell/scripts

步骤3.打开PuTTY/line命令对elastic1。

步骤4.运行命令：cd /intucell/scripts。

步骤5.运行命令：Python sc_eid_registration_prod.py csvfile.csv

Python sc_eid_registration_prod.py

```
import re
import subprocess
import sys
import csv
import datetime
import time
import os

def run(file_name):
    #import pdb;pdb.set_trace()
    print "Starting....."
    eids = readFromCSV(file_name)
    csvFilename="sc_eid_registration.csv"
    f = open('regEid.txt', 'w')
    for row in eids:
        header='Content-Type: application/xml'
        myURL="http://192.168.166.129:8083/pmg"
        #print "EID=%s"%eid
        myXml="<?xml version='1.0' encoding='UTF-8'?><Register
xmlns='http://www.cisco.com/ca/sse/PMGMessages-v2_0_0'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xsi:schemaLocation='http://www.cisco.com/ca/sse/PMGMessages-v2_0_0 pmg-messages-
v2_0_0.xsd'><TxnID>Register-TxnID-
0</TxnID><EID>%s</EID><Activated>>true</Activated><GroupMemberships><Group><Name>Israel</Name><Ty
pe>Area</Type></Group><Group><Name>%s</Name><Type>RFProfile</Type></Group></GroupMemberships></R
egister>"%(row['eid'],row['profile'])
        cmd='curl -X POST %s -vv -u "pmguser:pmguser" --digest -H "%s" -d
"%s"'%(myURL,header,myXml)
        cmd_ = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE)
        ok=False
        for line in cmd_.stdout:
```

```

        x = line.find("Success")
        if x!=-1:
            f.write("%s --- Register OK\n"%row['eid'])
            ok=True
            break
        if not ok:
            f.write("%s --- Register Fail\n"%row['eid'])
f.close() # you

def readFromCSV(csvFilename):
    eid_list=[]
    with open(csvFilename, "U") as f:
        reader = csv.DictReader(f)
        for line in reader:
            try:
                eid=str(line["EID"]).strip()
                profile=str(line["RFProfile_New"]).strip()
                if len(eid)>0 and len(profile)>0:
                    eid_list.append({'eid':eid,'profile':profile})
            except Exception as e:
                print "readFromCSV Error: %s"%e
                f.close()
                sys.exit()
    f.close()
    return eid_list

if __name__ == '__main__':
    try:
        usage="usage: python sc_eid_registration_prod.py <file name>"
        l=len(sys.argv)
        if l==2:
            run(sys.argv[1])
        else:
            print usage
    except Exception as e:
        print "ERROR:%"%e
        print usage
    finally:
        sys.exit()

```

当脚本完成时，报告文件自动地创建。

报告示例：

regEid.txt

```

001B67-352639055637721 --- Register OK
001B67-352639055637242 --- Register OK
001B67-352639055637218 --- Register OK
001B67-352639055637036 --- Register OK
001B67-352639055636947 --- Register OK
001B67-352639055636897 --- Register OK
001B67-352639055636830 --- Register OK
001B67-352639055636780 --- Register OK
001B67-352639055636764 --- Register OK
001B67-352639055636228 --- Register OK
001B67-352639055636137 --- Register OK
001B67-352639055635741 --- Register OK
001B67-352639055635295 --- Register OK
001B67-352639055635220 --- Register OK

```

001B67-352639055634959 --- Register OK
001B67-352639055633985 --- Register OK
001B67-352639055480304 --- Register OK
001B67-352639055480221 --- Register OK
001B67-352639055480130 --- Register OK
001B67-352639055480056 --- Register OK
001B67-352639055479785 --- Register OK
001B67-352639055479611 --- Register OK
001B67-352639055479546 --- Register OK
001B67-352639055479405 --- Register OK
001B67-352639055471162 --- Register OK
001B67-352639055470214 --- Register OK
001B67-352639055469539 --- Register OK
001B67-352639053871033 --- Register OK
001B67-352639053870704 --- Register OK
001B67-352639053863915 --- Register OK
001B67-352639053592746 --- Register OK
001B67-352639055781081 --- Register OK
001B67-352639055781073 --- Register OK
001B67-352639055781065 --- Register OK
001B67-352639055780877 --- Register OK
001B67-352639055780869 --- Register OK
001B67-352639055651912 --- Register OK
001B67-352639055651839 --- Register OK
001B67-352639055651789 --- Register OK
001B67-352639055651706 --- Register OK
001B67-352639055651672 --- Register OK
001B67-352639055651664 --- Register OK
001B67-352639055651656 --- Register OK

[验证](#)

当前没有可用于此配置的验证过程。

[故障排除](#)

目前没有针对此配置的故障排除信息。