

# Unidades novas do registro IMEI na pilha pequena universal de utilização do script automático RM 4.1 Python (USC)

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Verificar](#)

[Troubleshooting](#)

## Introdução

Este documento descreve o processo como de IMEI novo registrado nos serviços da gerência remotos (RM) que usam um script automático.

## Pré-requisitos

### Requisitos

Não existem requisitos específicos para este documento.

### [Componentes Utilizados](#)

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Informações de Apoio

Antes de executar este processo verifique por favor com a permissão, o usuário e a passagem do cliente ao server elástico remoto dos serviços da gerência (RM) (usuário de raiz).

No início do procedimento e na extremidade, verifique o relatório do uso do pool identificação para ver quanto o ceiling deixou.

o cellid duplicado pode ser criado na rede se não há bastante cellid no pool.

## Configurar

Execute o procedimento do registro:

Etapa 1. Crie o arquivo CSV de acordo com este formato, segundo as indicações da imagem:

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

Etapa 2. FTP o arquivo CSV ao server do elástico 1 RM com usuário/passagem

Biblioteca do lugar: **/intucell/scripts**

Etapa 3. Abra a massa de vidraceiro/linha de comando a elastic1.

Etapa 4. Execute o comando: **CD /intucell/scripts.**

Etapa 5. Execute o comando: **pitão sc\_eid\_registration\_prod.py csvfile.csv**

pitão 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()

```

O arquivo de relatório é criado automaticamente quando um script é feito.

Exemplo do relatório:

### 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

## Verificar

No momento, não há procedimento de verificação disponível para esta configuração.

## Troubleshooting

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.