

Nuevas IMEI unidades del registro en RM 4.1 usando célula universal del script automático de Python la pequeña (USC)

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Verificación](#)

[Troubleshooting](#)

Introducción

Este documento describe el proceso cómo de nuevo IMEI registrado en los servicios de administración remota (RM) usando un script automático.

Prerrequisitos

Requisitos

No hay requisitos específicos para este documento.

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

Antes de funcionar con este proceso marque por favor con el permiso, el usuario y el paso del cliente al servidor elástico de los servicios de administración remota (RM) (usuario raíz).

Al principio del procedimiento y en el extremo, marque el informe del uso del pool identificación para ver cuánto dejó el ceiling.

el cellid duplicado se puede crear en la red si no hay bastantes cellid en el pool.

Configurar

Funcione con el procedimiento del registro:

Paso 1. Cree archivo CSV según este formato, tal y como se muestra en de la imagen:

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

Paso 2. FTP archivo CSV al servidor elástico 1 RM con el usuario/el paso

Biblioteca de la ubicación: `/intucell/scripts`

Paso 3. Abra el putty/la línea de comando en elastic1.

Paso 4. Funcione con el comando: `/intucell/scripts cd`.

Paso 5. Funcione con el comando: `pitón sc_eid_registration_prod.py csvfile.csv`

pitón `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()

```

El archivo de informe se crea automáticamente cuando se hace un script.

Ejemplo del informe:

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

Verificación

Actualmente, no hay un procedimiento de verificación disponible para esta configuración.

Troubleshooting

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.