

Nouvelles IMEI unités du registre sur RMS 4.1 utilisant cellule universelle de script automatique de Python la petite (USC)

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurez](#)

[Vérifiez](#)

[Dépannez](#)

Introduction

Ce document décrit le processus comment de nouvel IMEI enregistré dans les services de gestion à distance (RMS) utilisant un script automatique.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

[Composants utilisés](#)

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

[Informations générales](#)

Avant qu'exécutant ce processus vérifiez s'il vous plaît avec l'autorisation, l'utilisateur et le passage de client au serveur élastique des services de gestion à distance (RMS) (utilisateur de base).

Au début de la procédure et à l'extrémité, vérifiez l'état d'utilisation de groupe d'id pour voir combien la décellulation a laissé.

le cellid en double peut être créé sur le réseau s'il n'y a pas assez de cellid sur le groupe.

Configurez

Exécutez la procédure de registre :

Étape 1. Créez le fichier de csv selon ce format, suivant les indications de l'image :

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

Étape 2. FTP le fichier de csv au serveur de l'élastique 1 RMS avec l'utilisateur/passage

Bibliothèque d'emplacement : **/intucell/scripts**

Étape 3. Ouvrez le mastic/ligne de commande à elastic1.

Étape 4. Exécutez la commande : **cd /intucell/scripts.**

Étape 5. Exécutez la commande : **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><Type>Area</Type></Group><Group><Name>%s</Name><Type>RFProfile</Type></Group></GroupMemberships></Register>"%(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()

```

Le fichier de rapport est automatiquement créé quand un script est fait.

Exemple d'état :

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

Vérifiez

Aucune procédure de vérification n'est disponible pour cette configuration.

Dépannez

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.