

## Manage Devices in CA PM

This document describes a sample Perl script to manage devices in CA PM from a CSV file in batch.

### Use Case

A customer extracts a list of devices (IP Address and host name by IP subnet) from a configuration database and would like to drive actions in CA PM such as

- add devices to new or existing discovery profile and optionally start the discovery
- remove devices from the discovery profile and delete the devices from CA PM
- change devices lifecycle state (set them to maintenance, retire or active)

### Approach

#### 1. Perl script manageDevices.pl

it reads configuration information and the host list from files and performs the given action via remote REST webservice calls

### Prerequisites

- Tested with CA PM 3.5 and 3.6
- Script host with Perl and access to CA PC port 8181 and CA DA port 8581
- Perl 5.18 or greater with module REST::Client

### Installation

- Unpack the archive on the script host to any directory
- We assume the Data Aggregator has CA PC DataSource ID 3. Please verify with following GET request:  
<http://CAPC:8181/pc/center/webservice/datasources/dataSourceId/3>
- if this is not the DA, find the right ID from <http://CAPC:8181/pc/center/webservice/datasources> and adjust the \$queryurl variable the script accordingly

### Script Usage

```
perl manageDevices.pl operation parameterFile ipFile [verbose level]
```

operation = one of the following:

*add* (add devices to discovery profile),

*addDiscover* (add devices to discovery profile and run the profile),

*remove* (remove devices from discovery profile and delete device from the DA),

*maintenance, retired, active* (change the lifecycle status for each device accordingly)

parameterFile = file with input parameters, e.g, CA PM system, tenant name, IPDomain name

ipFile = csv file with IP addresses (and hostnames)

also needed is DP\_template.xml: discovery profile template file with placeholders

verbose level = output messages to stdout: 0, 1 (default), 2, 3

## Examples

1. Setup parameter file (e.g. param.ini) file such as

```
CAPC=aos-capc:8181
CADA=aos-cada:8581
USER=Customer1_admin:secret
TENANT=Customer1
IPDOMAIN=customer1
PROFILENAME=
```

If the profilename is not explicitly specified here, it will be derived from the ipList filename (see below)

2. Provide the host list file (e.g. ipList103.csv). We assume the file looks like

```
IP,NAME
130.119.103.11,router-1
130.119.103.13,router-2
...
```

NOTE: the second column with the device name is informational only and not processed

3. Review the Discovery Profile template file: DP\_template.xml

```
<DiscoveryProfile version="1.0.0">
  <ActivationStatus>true</ActivationStatus>
  <IPListList>
  </IPListList>
  <RunStatus>READY</RunStatus>
  <Item version="1.0.0">
    <Name>__NAME__</Name>
  </Item>
  <IPDomainMember version="1.0.0">
    <IPDomainID>__IPDOMAINID__</IPDomainID>
  </IPDomainMember>
  <DeviceNameRankingList>
<DeviceNameRanking>{http://im.ca.com/inventory}ManageableDevice.SystemName</Dev
iceNameRanking>
<DeviceNameRanking>{http://im.ca.com/inventory}Device.HostName</DeviceNameRanki
ng>
<DeviceNameRanking>{http://im.ca.com/inventory}Device.PrimaryIPAddress</DeviceN
ameRanking>
  </DeviceNameRankingList>
<IcmpDiscoveryEnabled>true</IcmpDiscoveryEnabled>
</DiscoveryProfile>
```

\_\_NAME\_\_ and \_\_IPDOMAINID\_\_ are placeholders that will be substituted with the real value at runtime.

The \_\_NAME\_\_ placeholder can be surrounded by a constant prefix and suffix,

such as *DP\_\_NAME\_\_ Adresses*.

Edit this file if your name ranking should be different, or you need to assign specific SNMP profiles. Use an existing profile as a model and download it from <http://DA:8581/rest/discoveryprofiles>

4. Add devices and run discovery, first run for this ipList file

```
perl manageDevices.pl addDiscover param.ini ipList103.csv
```

output – a new discovery profile named *ipList103* is newly created

```
profilename=ipList103
tenantID=31970
IPdomainID=32006
profileID created=58199
```

```
profile run rc=200
```

#### 5. Add more devices and run discovery

The provisioning system has added additional entries to ipList103.csv. Another run of the script will just add the new entries to the (existing) profile and run it.

```
perl manageDevices.pl addDiscover param.ini ipList103.csv
```

**output** – it shows the discovery profile already exists

```
profilename=ipList103
tenantID=31970
IPdomainID=32006
profileID found=58199
add to profile rc=200
profile run rc=200
```

#### 6. Remove devices

The provisioning system has recreated ipList103.csv with decommissioned devices, they should be deleted from the system and removed from the discovery profile

```
perl manageDevices.pl remove param.ini ipList103.csv
```

**output:**

```
profilename=ipList103
tenantID=31970
IPdomainID=32006
profileID found=58199
remove from profile rc=200
devices remove request rc=200
delete devices rc=200
```

#### 7. Set devices to maintenance

The provisioning system has recreated ipList103.csv with devices that should be set to maintenance mode

```
perl manageDevices.pl maintenance param.ini ipList103.csv
```

**output:**

```
profilename=ipList103
tenantID=31970
IPdomainID=32006
profileID found=58199
change state for PC item 26465 rc=200
```

#### 8. Use case: specific format for the profile name

For example, the discovery profile names should follow the format “x.y.z Addresses”

- a. Edit DP\_template.xml, the <Name> tag should read

```
<Name>__NAME__ Addresses</Name>
```

- b. The files with the IP addresses are named like 10.31.17.csv

c. Run script

```
perl manageDevices.pl addDiscover param.ini 10.31.17.csv  
for the first time with that input file will create the discovery profile  
10.31.17 Addresses
```

You can verify in script output

```
profilename=10.31.17 Addresses
```

...

## Notes

1. CA PC User name and password

The CA PC credentials are only used when changing the lifecycle state (maintenance, retire, active). You may specify global admin account or tenant admin account.

2. Verbose level

Verbose level is the optional fourth parameter and controls the output messages from the REST calls

0: not output

1 (default): item IDs illustrating sequence of steps, some returncodes

2: all returncodes and some response content

3: all response content and payloads

3. Understand returncodes

rc=200 success

rc=400 may happen e.g. if trying to delete a not existing device or activating an active device

4. Known Issue

When trying to re-add IP addresses that **all** have previously been removed, they are not added to the discovery profile. This is a known issue of the DA REST webservice. If also new IP addresses are among that add request, there should not be an issue.

Current workaround: add missing IP addresses manually to the profile.