

# How to Generate Provisioning Files for Many Phones

This tutorial will help you generate provisioning files for many phones using SystemAPI.

- [Why generate provisioning files](#)
- [Requirements](#)
- [Setup](#)
- [How it works](#)
- [Tips and tricks](#)

## Why generate provisioning files

By using VoipNow's provisioning feature, you can set and maintain identical configurations for a large amount of equipment.

Suppose your office uses the same phone model for all employees. Automating the generation of provisioning files through SystemAPI would be an easy and elegant method to configure those phones.

Let's say you are the network administrator of an office with 15 employees. You're supposed to configure 15 Cisco terminals for them. Wouldn't it be nice to use a single script to generate provisioning files for all those phones?

You could very easily do that if you follow the steps below. Additionally, a report with the configuration link for every phone will be generated for your own use.

## Requirements

To generate bulk provisioning files for multiple phones using this example you need:

- An access token; add an app to the VoipNow Platform following [these instructions](#), then follow the steps described [here](#) to obtain the access token;
- Set the provisioning system IP as detailed [here](#);
- A Phone Terminal extension for each employee;
- The MAC address of every phone you want to provision;
- The country code of your office.

## Setup

Download the [example attached](#) and follow the steps below.

1. Configure the connection and Auth details. To make a request to the VoipNow Platform API, you need to configure a few variables:
  - **\$IP**: IP of the VoipNow server (including port);
  - **\$VERSION**: Version of the VoipNow product;
  - **\$TOKEN**: Authentication token;

```
/* IP of the VoipNow server (including port) */
$IP = "192.168.14.244"; //CHANGE
/* Version of the VoipNow product */
$VERSION = '3.0.0'; //CHANGE
/* Authentication token*/
$TOKEN= '1|y_J5vqVg4D.8qORSXbT-qnK3GHVB_XOM|1|1365672971|0-t01mDDGG_BXGJ8nwQ-wBMWxtjU-QO2'; //CHANGE
```

2. Set the map of extensions to phone MAC addresses. The map can be kept in an array that uses the extension number as key and the phone's MAC address as value.

```
/* Create an array having as key the extension number and as value the MAC address of each phone terminal */
$mac_array = array(
    '0003*001' => '00:18:B9:AA:BB:01',
    '0003*002' => '00:18:B9:AA:BB:02',
    '0003*003' => '00:18:B9:AA:BB:03'
);
```

3. Configure the country code. Assuming your office is located in Rome, Italy, you should set \$country\_code to IT. The script will use the SystemAPI GetTimezone method to find the timezone ID.

```
$country_code='IT';
```

- Set the phone model. Perform a SystemAPI request to get the phone model. Assuming you have to configure *Cisco 7940* phones with firmware version 8.x, you should perform a request to `GetEquipmentList` method and obtain the required data from the response.

```
$phone_model = X //Equipment unique ID
$firmware = Y //Firmware unique ID
$protocol = Z //Firmware protocol
```

Here is a snapshot from the `GetEquipmentList` response which contains explanations on how to get the required data according to the phone model:

```
<!-- The response of GetEquipmentList SystemAPI method for a phone model: -->
...
<ns2:uniqueID>gmf6izchfvd14fg4xwclpz9kv20t8eko</ns2:uniqueID> <!-- This is the value for X in our example -->
<ns2:model>Cisco 7940</ns2:model>
<ns2:firmware>
  <ns2:uniqueID>sq8uondpcyzcf776sev7sv6vt8o82klu</ns2:uniqueID>
  <ns2:name>7.x</ns2:name>
  <ns2:protocol>TFTP</ns2:protocol>
</ns2:firmware>
<ns2:firmware>
  <ns2:uniqueID>a9pzw3o92omhwtno7jwzf2v8ajhd3j8e</ns2:uniqueID> <!-- This is the value for Y in our example -->
  <ns2:name>8.x</ns2:name>
  <ns2:protocol>TFTP</ns2:protocol> <!-- This is the value for Z in our example -->
</ns2:firmware>
...
```

## How it works

- Using the connection and auth details provided, create and authenticate the `SoapClient`.
- Using the `country_code` variable, retrieve the needed timezone ID. Here is a code example of how to obtain the timezone ID.

```
//example how to get timezone id

$country_code='IT'; // country code for Italy is IT
try {
    $request_timezone = array(
        'code' => $country_code
    );

    $timezone_result = $client->GetTimezone($request_timezone); // call SystemAPI GetTimezone
method
    $timezone_id = $timezone_result->timezone->ID;
} catch (SoapFault $exception) {
    echo "ERROR: <br/>" . $exception->getMessage() . "<br/><br/>";
    echo "REQUEST:<br/>" . htmlentities($client->__getLastRequest()) . "<br/><br/>";
    echo "RESPONSE:<br/>" . htmlentities($client->__getLastResponse()) . "<br/>";
}
```

- Get the phone model, firmware, and protocol. Here is a code example of how to parse `GetEquipmentList` response.

```

/* Use GetEquipmentList method to find the code for your phone:*/
try {
    $equipment_list_result = $client->GetEquipmentList();

    /* Get an array of all equipments */
    $all_phones = $equipment_list_result->equipment;

    /* The data for Cisco 7940 phones */
    $cisco_phones = (array)$all_phones[6];

    /* Get Phone Model, Firmware and Protocol for model 8.x */
    $phone_model = $cisco_phones['uniqueID'];
    $firmware = $cisco_phones['firmware'][1]->uniqueID;
    $protocol = $cisco_phones['firmware'][1]->protocol;

} catch (SoapFault $exception) {
    echo "ERROR: <br/>" . $exception->getMessage() . "<br/><br/>";
    echo "REQUEST:<br/>" . htmlentities($client->__getLastRequest()) . "<br/><br/>";
    echo "RESPONSE:<br/>" . htmlentities($client->__getLastResponse()) . "<br/>";
}

```

4. Generate the provisioning files. Iterate through each extension & MAC pair and generate the provisioning files for each device using SetProvision SystemAPI method.

```

try {
    $request_provision = array(
        'extendedNumber' => $extension, //The provisioned extension
        'provision' => true, //Use provisioning
        'name' => 'provision for extension '.$extension, //Friendly name
        'phoneModel' => $phone_model, //Manufacturer
        'firmware' => $firmware, //Model
        'protocol' => $protocol, //Update protocol
        'phoneMAC' => $mac, //MAC address
        'timezoneID' => $timezone_id, //Phone time zone
        'connectionType' => 'dhcp', //phones connect using DHCP
    );

    $provisioning_result = $client->SetProvision($request_provision);

    $file = $provisioning_result->location; //The provisioning file address

} catch (SoapFault $exception) {
    echo "ERROR: <br/>" . $exception->getMessage() . "<br/><br/>";
    echo "REQUEST:<br/>" . htmlentities($client->__getLastRequest()) . "<br/><br/>";
    echo "RESPONSE:<br/>" . htmlentities($client->__getLastResponse()) . "<br/>";
}

```

5. Display the links for the locations of the provisioning files. Once the script has run a page listing each extension, the provisioning file is created. The list looks as shown below.

Extension	MAC	File
0003*001	00:18:B9:AA:BB:01	TFTP://192.168.14.244/pro/p/ld/3fc038628994/SIP0018B9AABB01.cnf
0003*002	00:18:B9:AA:BB:02	TFTP://192.168.14.244/pro/p/ld/5b0313842006/SIP0018B9AABB02.cnf
0003*003	00:18:B9:AA:BB:03	TFTP://192.168.14.244/pro/p/ld/fac272556511/SIP0018B9AABB03.cnf
0003*004	00:18:B9:AA:BB:04	TFTP://192.168.14.244/pro/p/ld/2e4731972893/SIP0018B9AABB04.cnf

This is how the report page is created.

```

....
/* Construct the content of report page */
$html = '<table border="1">
    <tr>
        <th>Extension</th>
        <th>MAC</th>
        <th>File</th>
    </tr>
';

....
/* Create a row in table for each provisioned extension */
$html .= '<tr>
    <td>' . $extension . '</td>
    <td>' . $mac . '</td>
    <td>' . $file . '</td>
</tr>';

....
$html .= '</table>';

/* Write the content of $html variable in a file. Create a link towards this file */
$handle = fopen("phones_provisioning_files.html", "w");
fwrite($handle, $html);
fclose($handle);
echo "The report was successful generated. You can find it <a href=\"http://\" . $_SERVER["SERVER_ADDR"] . "/" .
dirname($_SERVER["SCRIPT_NAME"]) . "/phones_provisioning_files.html\">here</a>";

```

## Tips and tricks

The country code is used to get the timezone ID so that the provisioned phones display the hour accurately.

Need Help? Ask a question in our [GetSatisfaction](#) community.