SMART/RG®
forward thinking

501 SE Columbia Shores Boulevard, Suite 500
Vancouver, Washington, 98661 USA
+360 859 1780 / smartrg.com

/DEVICE MANAGER API ESSENTIALS

Common Life Cycle

Version 2.1.2

January 2019

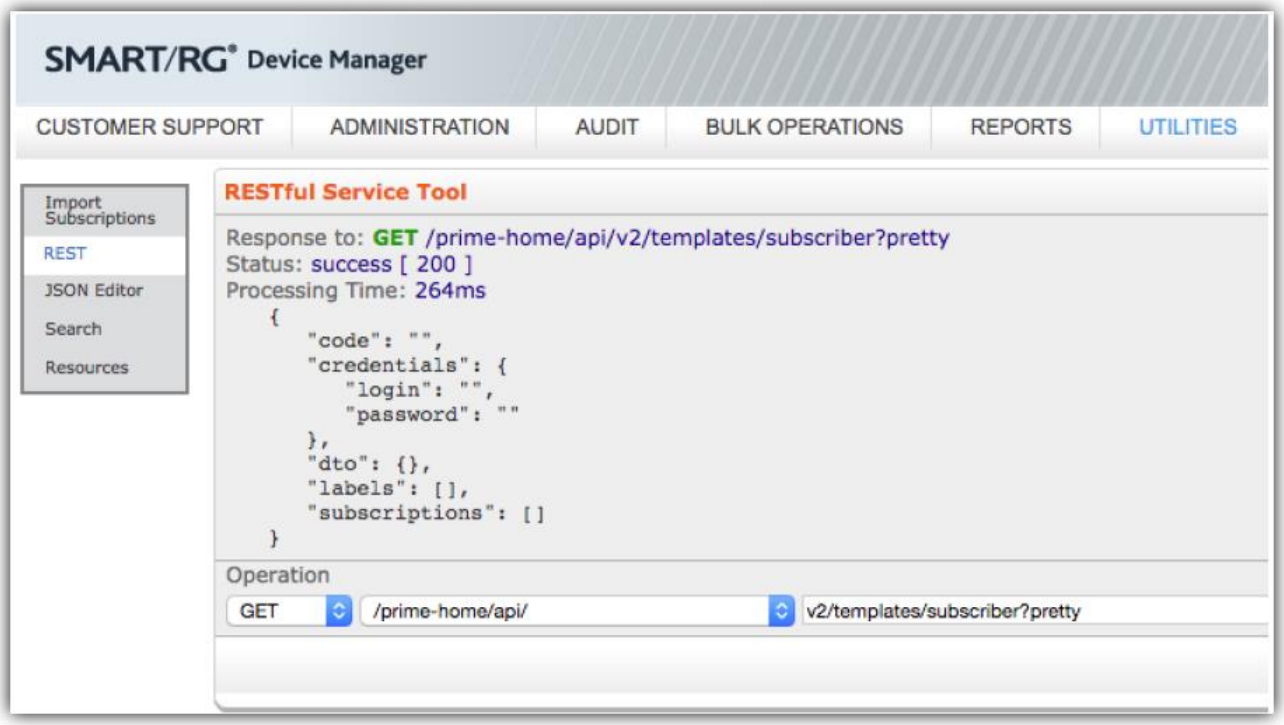
SMART/RG®
forward thinking

PURPOSE & SCOPE / OVERVIEW

The purpose of this document is to present the typical Device Manager ACS subscriber/device record life cycle. It is IMPORTANT TO NOTE the examples presented here represent a small subset of the overall capability of the Device Manager API. See the "Device Manager by SmartRG Integration Guide" for a full description of the Device Manager API.

PROTOTYPING API CALLS – RESTFUL SERVICES TOOL

The Device Manager ACS includes built in tools to make API integration easy. The RESTful Services tool (found under UTILITIES) enables developers to rapidly prototype API calls before coding begins. Should you encounter API development challenges, the RESTful Services tool is a powerful troubleshooting and debugging resource.



SUBSCRIBER/DEVICE RECORD LIFECYCLE

The typical a subscriber/device record lifecycle in the SmartRG Device Manager ACS is:

- Create a new subscriber/device record
  - GET a device template
  - Modify and POST the updated device template to create the device
  - GET a subscriber template
  - Modify and POST the updated subscriber template to create the subscriber
  - GET the device actions
  - Modify the device actions to force an outbound sync of the new settings
- Modify an existing subscriber/device record's settings

- Query the subscriber based on subscriber code (account number)
- GET the subscriber record using the subscriberId from the query
- Modify and PUT the updated subscriber record
- GET the device actions
- Modify the device actions to force an outbound sync of the new settings

- DELETE an existing subscriber/device record
  - Query the subscriber record based on subscriber code (account number)
  - DELETE the device record using the deviceId from the subscriber record
  - DELETE the subscriber record using the subscriberId from the subscriber record

Create a New Subscriber/Device Record

The following resource endpoints and JSON data structures shall be used to create new subscriber/device records.

Get Subscriber Template

HTTP Command: GET

URL: http://ACSName.smartrg.com/prime-home/api/v2/templates/subscriber

Response:

```
{
    "dto": {},
    "subscriptions": [],
    "labels": [],
    "credentials": {
        "login": "",
        "password": ""
    },
    "code": ""
}
```

Create Subscriber

Modify the subscriber template returned in the previous step to include subscriber attributes, subscriptions and settings and POST the updated subscriber data.

IMPORTANT NOTE: If you intend to use *Control Panel Provisioning*, see the highlighted notes in the Request JSON data structure below.

HTTP Command: POST

URL: http://ACSName.smartrg.com/prime-home/api/v1/subscribers/

Request JSON:

```
{
    "dto": {},
    "subscriptions": [
        {
            // If you are NOT using Control Panel Provisioning, use the following "device" data structure
            "device": {
                "deviceId": 69561,
                "sn": "3C906618CBE0",
                "oui": "3C9066",
                "labels": []
            },
            // If you ARE using Control Panel Provisioning, use the following "device" data structure instead
            "device": {
                "cpProvisioningId":"1",
                "disposition":"FUTURE_DEVICE",
                "labels":[]
            },
            "settings": {
                "Settings.WANPPPConnections.1.Username": "uname@myISP.net",
                "Settings.WANPPPConnections.1.Password": "TYhpXsK76s77",
                "Settings.WIFI.Radio.1.ChannelBandwidth": "20MHz",
                "Settings.WIFI.Radio.1.FrequencyBand": "2.4GHz",
                "Settings.WIFI.Radio.1.OperatingStandards": "n",
                "Settings.WIFI.Radio.1.Channel": "3",
                "Settings.WIFI.Radio.2.ChannelBandwidth": "20MHz",
```

```
                "Settings.WIFI.Radio.2.FrequencyBand": "5GHz",
                "Settings.WIFI.Radio.2.OperatingStandards": "ac",
                "Settings.WIFI.Radio.2.Channel": "140",
                "Settings.WIFI.AccessPoint.1.EnableLocalNetworkAccess":
"true",
                "Settings.WIFI.AccessPoint.1.EnableAdvertisement": "true",
                "Settings.WIFI.AccessPoint.1.SSID": "BigBird-24GHz",
                "Settings.WIFI.AccessPoint.1.Radios.1.Reference": "1",
                "Settings.WIFI.AccessPoint.1.Enable": "true",
                "Settings.WIFI.AccessPoint.2.EnableLocalNetworkAccess":
"true",
                "Settings.WIFI.AccessPoint.2.EnableAdvertisement": "true",
                "Settings.WIFI.AccessPoint.2.SSID": "BigBird-5GHz",
                "Settings.WIFI.AccessPoint.2.Radios.1.Reference": "2",
                "Settings.WIFI.AccessPoint.2.Enable": "true",
                "Settings.WIFI.AccessPoint.1.SecurityMode": "WPA2-
Personal",
                "Settings.WIFI.AccessPoint.1.Secret": "password123",
                "Settings.WIFI.AccessPoint.2.SecurityMode": "WPA2-
Personal",
                "Settings.WIFI.AccessPoint.2.Secret": "password123"
            },
            "services": [
                {
                    "status": "DISABLED",
                    "name": "Content Filtering"
                },
                {
                    "status": "DISABLED",
                    "name": "Time Blocking"
                },
                {
                    "status": "ENABLED",
                    "name": "Wireless"
                },
                {
                    "status": "DISABLED",
```

```
                    "name": "IPTV"
                }
            ]
        }
    ],
    "labels": [],
    "credentials": {
        "login": "bigyellowbird",
        "password": "TYhpXsK76s77"
    },
    "code": "212-555-1234",
    "attributes": {
        "Subscriber.Address.1.City": "New York",
        "Subscriber.Address.1.PostalCode": "10004",
        "Subscriber.Address.1.State": "NY",
        "Subscriber.Address.1.Street": "123 Sesame St.",
        "Subscriber.Address.1.Type": "Home",
        "Subscriber.EmailAddress": "myEmail@myISP.net",
        "Subscriber.FullName": "Big Bird",
        "Subscriber.Phone.1.Number": "212-555-1234",
        "Subscriber.Phone.1.Type": "Home"
    }
}
```


GET Device Template:

IMPORTANT NOTE: If you intend to use *Control Panel Provisioning* (where the device is serial number is unknown until the first Control Panel login), skip the device template GET and POST steps. See the "Create Subscriber/Device Record Logic Diagram" for details.

HTTP Command: GET

URL: http://ACSName.smartrg.com/prime-home/api/v1/templates/device

Response:


```
{
    "actionLog": [],
    "labels": [],
```

```
    "applications": {
        "FirmwareManagement": {
            "dto": {}
        },
…
        "WIFI": {
            "dto": {}
        },
…
        "CAPO": {
            "dto": {}
        },
    },
    "queuedActions": {
        "scripts": [],
        "services": {
            "PortForwards": {
                "canToggleStatus": false,
                "status": "ENABLED"
            },
            "firmware": {
                "canToggleStatus": false,
                "status": "ENABLED"
            },
…
        },
        "applications": {
            "ManagementServerStatus": {
                "pendingSync": false,
                "dataOwner": "DEVICE"
            },
            "Hosts": {
                "pendingSync": false,
                "dataOwner": "DEVICE"
            },
…
```

```
        }
    }
}
```

NOTE: The JSON data structure above has been truncated for brevity. To view the entire data structure use the Device Manager RESTful Services tool (under UTILITIES) and enter the device template GET URL to see a complete list of all applications and services returned in a device template JSON data structure.

Create Device

Modify the device template returned in the previous step to include the:

- subscriberCode (shown as “code” in the subscriber JSON data structure above)
- device OUI
- deviceOUI

and then POST the updated structure to create the device. See the highlighted values below.

NOTE: subscriberCode is an optional value but when supplied in this API call, there needs to be a subscriber in Device Manager with that subscriberCode.

HTTP Command: POST

URL: http://ACSName.smartrg.com/prime-home/api/v1/devices

Request JSON:

```
{
    "actionLog": [],
    "labels": [],
    "applications": {
        "FirmwareManagement": {
            "dto": {}
        },
…
    },
    "queuedActions": {
        "scripts": [],
        "services": {
            "PortForwards": {
                "canToggleStatus": false,
                "status": "ENABLED"
```

```
            },
            "firmware": {
                "canToggleStatus": false,
                "status": "ENABLED"
            },
…
    },
    "sn": "3C906618CBE0",
    "oui": "3C9066"
}
```

Store the deviceId received in the device create POST response to be used in the subscriber create step to follow.

Use API to trigger a device to synchronize specific applications

ACS API Endpoint: `http(s)://<ACS URL>/prime-home/api/v1/devices/<Device ID #>/actions`
Example: `http(s)://qadm10.smartrg.com/prime-home/api/v1/devices/12321/actions`

Use Case

The 'actions' API endpoint can be used to mark an application or script for synchronization with the device in question. This can be used by third party applications to pull required data into the ACS for further analysis, or it can be use by third party applications to push modified device data down to the device in question. (Only applications are shown in the example)

NOTE: All API calls are asynchronous, meaning that they will return their status code regardless of whether or not the device has been solicited and data has been collected from or set to the device in question.

Summarized Steps

1. Perform a GET on the URL above to retrieve the devices actions
2. Modify the JSON data so that the intended application has the relevant settings below
    a. Application that can PUSH and PULL (WiFi, Port Forwards, etc.)
        i. PULL: "needsInitialization": true
        ii. PUSH: "pendingSync": true
    b. Application that can ONLY PUSH or PULL (LAN Hosts, Network Time, etc.)
        i. "pendingSync": true
3. If the device needs to be actively solicited
    a. Modify the JSON by adding the following entry in the root of the JSON structure
        i. "solicit": true
        ii. See <Some Example>
4. Perform a PUT to the same API endpoint with the modified JSON

Detailed Steps Explained

Step 1: GET the 'actions' end point to retrieve the JSON data for the desired device

HTTP Request Command: GET
JSON Response

```
{
  "solicitableStatus": "OK",
  "revision": -1104981854,
  "scripts": [
    ......
  ],
  "services": {
    ......
  },
  "refreshScript": [
    ......
  ],
  "applications": {
    "Hosts": {                  // PULL Only Application
      "pendingSync": false,
      "dataOwner": "DEVICE"
    },
    "Time": {                   // PUSH Only Application
      "pendingSync": false,
      "dataOwner": "SERVER"
    },
    "WIFI": {                   // PUSH and PULL Application
      "needsInitialization": false,
      "pendingSync": false,
      "dataOwner": "SERVER"
    },
    "PF": {                     // PUSH and PULL Application
      "needsInitialization": false,
      "pendingSync": false,
      "dataOwner": "SERVER"
    },
    ......
  }
}
```

Step 2: Modify Response JSON to fit desired application behavior

```
{
  "solicitableStatus": "OK",
  "revision": -1104981854,
  "scripts": [
    ......
  ],
  "services": {
    ......
  },
  "refreshScript": [
    ......
  ],
  "applications": {
    "Hosts": {                        // PULL Only Application
      "pendingSync": true,            // Set to true to PULL Hosts
      "dataOwner": "DEVICE"
    },
    "Time": {                         // PUSH Only Application
      "pendingSync": true,            // Set to true to PUSH Time
      "dataOwner": "SERVER"
    },
    "WIFI": {                         // PUSH and PULL Application
      "needsInitialization": true,  // Set to true to PULL WiFi
      "pendingSync": true, // Set to true to mark for synchronization
      "dataOwner": "SERVER"
    },
    "PF": {                       // PUSH and PULL Application
      "needsInitialization": false,
      "pendingSync": true,      // Set to true to PUSH Port Forwards
      "dataOwner": "SERVER"
    },
    ......
  }
  "Solicit": true    // OPTIONAL Set to true to solicit the device
}
```

Step 3: PUT the modified JSON back to the 'actions' endpoint

HTTP Request Command: PUT
Modified JSON:

```
{
  "solicitableStatus": "OK",
  "revision": -1104981854,
  "scripts": [
    ......
  ],
  "services": {
    ......
  },
  "refreshScript": [
    ......
  ],
  "applications": {
    "Hosts": {                              // PULL Only Application
      "pendingSync": true,
      "dataOwner": "DEVICE"
    },
    "Time": {                               // PUSH Only Application
      "pendingSync": true,
      "dataOwner": "SERVER"
    },
    "WIFI": {                               // PUSH and PULL Application
      "needsInitialization": true,
      "pendingSync": false,
      "dataOwner": "SERVER"
    },
    "PF": {                                 // PUSH and PULL Application
      "needsInitialization": false,
      "pendingSync": false,
      "dataOwner": "SERVER"
    },
    ......
  }
  "Solicit": true
}
```

Use API to check if applications have synchronized

ACS API Endpoint: `http(s)://<ACS URL>/prime-home/api/v1/devices/<Device ID #>/actions/status`
Example: `http(s)://qadm10.smartrg.com/prime-home/api/v1/devices/12321/actions/status`

Summarized Steps (using Synchronization above)

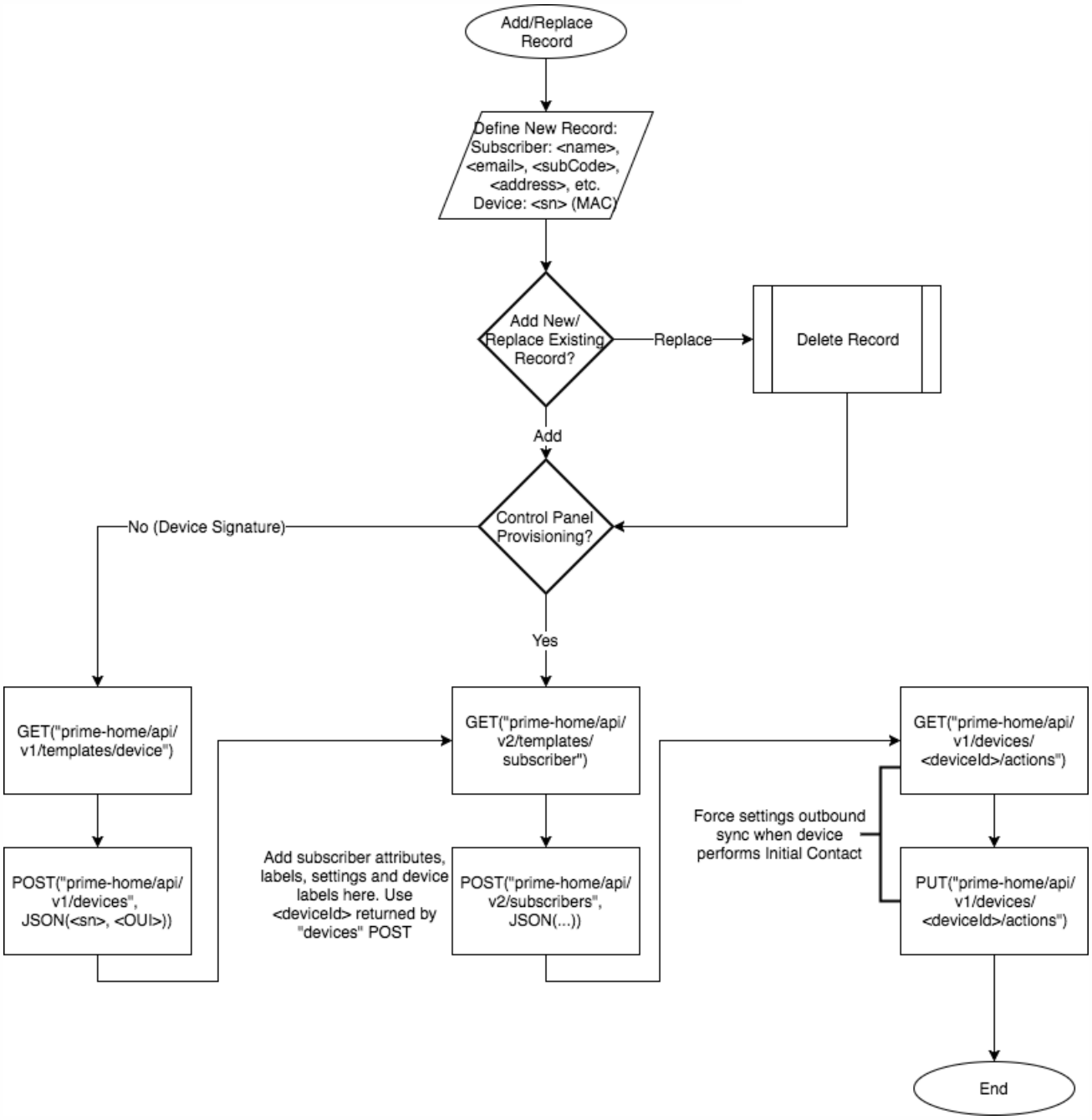
1. Build URL that contains the '*since*' time and the apps that have been set to sync.
    a. since=<Time in MS>
        i. This is the time at which the ACS will check each for the status of each application
    b. pendingSyncs=<Application Code>
        i. This parameter must be entered for each application that is expected to synchronize
    c. Example
        i. `actions/status?since=1547501412000&pendingSyncs=WANStatus&pendingSyncs=Hosts`

HTTP Request Command: GET
Example For Above Solicit: http(s)://qadm10.smartrg.com/prime-home/api/v1/devices/12321/actions/status?since=1547501412000&pendingSyncs=Hosts&pendingSyncs=Time&pendingSyncs=WiFi&pendingSyncs=PF

HTTP Response

```
{
  "queuedRuns": [],
  "syncApplications": [
    {
      "complete": true,
      "appCode": "Hosts"
    },
    {
      "complete": false,
      "appCode": "Time"
    },
    {
      "complete": false,
      "appCode": "WiFi"
    },
    {
      "complete": false,
      "appCode": "PF"
    }
  ],
  "updatedServices": [],
  "completed": false
}
```

Create Subscriber/Device Record Logic Diagram



Modify an Existing Subscriber/Device Record

The following resource endpoints and JSON data structures shall be used to modify an existing subscriber/device record.

Query the Subscriber

In order to modify an existing record it first must be found and retrieved from the ACS database. Subscribers are referenced by a unique identifier known as the "Subscriber Code." Often the Subscriber Code is synonymous with the billing system account number.

To find an existing subscriber POST a query to search for the record containing the specified Subscriber Code.

Note: Records can be identified by any unique attribute including: name, email address, etc. See the "Device Manager by SmartRG Integration Guide" for a full description of query types.

HTTP Command: POST

URL: http://ACSName.smartrg.com/prime-home/portal/query/execute

Request Data: code: mySubscriberCode

Response JSON:


```
{
    "id": "3551#6321",
    "docId": "subscription#3551#6321",
    "type": "subscription",
    "fields": {
      "wanType": [
        "Bonded VDSL2"
      ],
      "model": "SR552n - SR552n",
      "emailAddress": "myEmail@myISP.com",
      "serialNumber": "00236A5BE75F",
      "wanIPv4Address": [
        "075.175.119.172"
      ],
      "lastInform": "2016-11-11T16:31:24.364Z",
      "subscriberCode": "mySubscriberCode",
      "oui": "00236A",
      "manufacturer": "SmartRG",
      "subscriberId": "3551",
      "fullName": "Big Y. Bird",
      "deviceId": "6321",
```

```
      "disposition": "MANAGED_DEVICE"
    }
  },
  {
    "id": "3551",
    "docId": "subscriber#3551",
    "type": "subscriber",
    "fields": {
      "subscriberCode": "mySubscriberCode",
      "emailAddress": "myEmail@myISP.com",
      "fullName": "Big Y. Bird"
    }
  }
```


GET the Subscriber Record

Subscriber records are identified by a unique handle known as a "subscriberId." The subscriberId needed to reference a subscriber's settings is returned in the query above. See the highlighted subscriberId.

HTTP Command: GET

URL: http://ACSName.smartrg.com/prime-home/api/v1/subscribers/3551

Response JSON:


```
{
    "attributes": {
      "Subscriber.EmailAddress": "myEmail@myISP.com",
      "Subscriber.FullName": "Big Y. Bird"
    },
    "code": "mySubscriberCode",
    "credentials": {
      "expires": 1478634295462,
      "login": "bigYellowBird"
    },
    "labels": [],
    "revision": -751942221,
    "subscriberId": 3551,
    "subscriptions": [
      {
```

```
        "device": {
          "deviceId": 6321,
          "labels": [],
          "oui": "00236A",
          "sn": "00236A5BE75F"
        },
        "services": [
          {
            "name": "Circle",
            "status": "DISABLED"
          },
          {
            "name": "Sync After Copy",
            "status": "DISABLED"
          },
          {
            "name": "ICMP",
            "status": "DISABLED"
          },
          {
            "name": "WiFi Test",
            "status": "DISABLED"
          }
        ],
        "settings": {
          "Settings.AccessControlStatus.FTP.LAN.Enabled": "false",
          "Settings.AccessControlStatus.FTP.LAN.Port": "21",
          "Settings.AccessControlStatus.FTP.WAN.Enabled": "false",
          "Settings.AccessControlStatus.FTP.WAN.Port": "21",
          "Settings.AccessControlStatus.HTTP.LAN.Enabled": "true",
          "Settings.AccessControlStatus.HTTP.LAN.Port": "80",
…
          "Settings.DHCPStaticReservations.1.IPAddress": "192.168.1.9",
          "Settings.DHCPStaticReservations.1.MACAddress":
"14:10:9f:cf:0a:4b",
```

```
          "Settings.DHCPStaticReservations.1.Managed.1.ManagedBy":
"Settings.StaticReservations.1",
          "Settings.DHCPStaticReservations.1.Managed.2.ManagedBy":
"Settings.StaticReservations.2",
          "Settings.DeviceInfo.DSL.1.AttenuationDown": "32.3",
          "Settings.DeviceInfo.DSL.1.AttenuationUp": "0",
          "Settings.DeviceInfo.DSL.1.CurrentRateDown": "10062",
          "Settings.DeviceInfo.DSL.1.CurrentRateUp": "941",
          "Settings.DeviceInfo.DSL.1.MaxBitRateDown": "26951000",
          "Settings.DeviceInfo.DSL.1.MaxBitRateUp": "949000",
…
          "Settings.Hosts.1.Active": "true",
          "Settings.Hosts.1.Alias": "bigbird-VAIO",
          "Settings.Hosts.1.IPv4Addresses.1.IPv4Address": "192.168.1.10",
          "Settings.Hosts.1.Icon": "laptop.png",
          "Settings.Hosts.1.InterfaceType": "802.11",
          "Settings.Hosts.1.MACAddress": "00:23:15:51:27:48",
          "Settings.Hosts.1.Manufacturer": "Intel Corporate",
          "Settings.Hosts.1.Wlan.RateFromHost": "144444",
          "Settings.Hosts.1.Wlan.RateToHost": "144444",
          "Settings.Hosts.1.Wlan.Strength": "-31",
…
          "Settings.LAN2.DHCP.MaxAddress": "192.168.1.254",
          "Settings.LAN2.DHCP.MinAddress": "192.168.1.2",
          "Settings.LAN2.DHCP.ServerEnable": "1",
          "Settings.LAN2.DHCP.SubnetMask": "255.255.255.0",
          "Settings.LAN2.IPInterface.1.DHCPGateway": "true",
          "Settings.LAN2.IPInterface.1.Enable": "true",
          "Settings.LAN2.IPInterface.1.IPAddress": "192.168.1.1",
          "Settings.LAN2.IPInterface.1.SubnetMask": "255.255.255.0",
          "Settings.ManagementServerStatus.ConnectionRequestURL":
"http://75.175.119.172:30005/",
          "Settings.ManagementServerStatus.ConnectionRequestUsername":
"admin00236A5BE75F",
          "Settings.ManagementServerStatus.PeriodicInformInterval":
"82800",
          "Settings.ManagementServerStatus.UDPConnectionRequestAddress":
"",
```

```
            "Settings.PortForwards.1.Description": "Teredo",
            "Settings.PortForwards.1.EndPort": "59634",
            "Settings.PortForwards.1.IPAddress": "192.168.1.16",
            "Settings.PortForwards.1.InternalPort": "59634",
            "Settings.PortForwards.1.Protocol": "UDP",
            "Settings.PortForwards.1.StartPort": "59634",
            "Settings.PortForwards.2.Description": "Skype UDP at
192.168.1.14:5634 (3602)",
            "Settings.PortForwards.2.EndPort": "5634",
            "Settings.PortForwards.2.IPAddress": "192.168.1.14",
            "Settings.PortForwards.2.InternalPort": "5634",
            "Settings.PortForwards.2.Protocol": "UDP",
            "Settings.PortForwards.2.StartPort": "5634",
…

"Settings.WANStatus.Interfaces.1.IPv4Addresses.1.AddressingType": "IPCP",
            "Settings.WANStatus.Interfaces.1.IPv4Addresses.1.IPv4Address":
"75.175.119.172",
            "Settings.WANStatus.Interfaces.1.LANBridgeEnabled": "false",
            "Settings.WANStatus.Interfaces.1.LinkVLANId": "201",
            "Settings.WANStatus.Interfaces.1.TR069Paths.1.TR069Path":
"InternetGatewayDevice.WANDevice.2",
            "Settings.WANStatus.Interfaces.1.TR069Paths.2.TR069Path":
"InternetGatewayDevice.WANDevice.13",
            "Settings.WANStatus.Interfaces.1.TR069Paths.3.TR069Path":
"InternetGatewayDevice.WANDevice.2.WANDSLInterfaceConfig",
            "Settings.WANStatus.Interfaces.1.TR069Paths.4.TR069Path":
"InternetGatewayDevice.WANDevice.13.WANDSLInterfaceConfig",
            "Settings.WANStatus.Interfaces.1.TR069Paths.5.TR069Path":
"InternetGatewayDevice.WANDevice.2.WANConnectionDevice.2.WANPTMLinkConfig"
,
            "Settings.WANStatus.Interfaces.1.TR069Paths.6.TR069Path":
"InternetGatewayDevice.WANDevice.2.WANConnectionDevice.2.WANPPPConnection.
1",
            "Settings.WANStatus.Interfaces.1.WANType": "Bonded VDSL2",
            "Settings.WIFI.AccessPoint.1.Enable": "true",
            "Settings.WIFI.AccessPoint.1.EnableAdvertisement": "true",
            "Settings.WIFI.AccessPoint.1.EnableLocalNetworkAccess": "true",
            "Settings.WIFI.AccessPoint.1.Radios.1.Reference": "1",
```

```
            "Settings.WIFI.AccessPoint.1.SSID": "bigbirdWiFi",
            "Settings.WIFI.AccessPoint.1.Secret": "burtandernie",
            "Settings.WIFI.AccessPoint.1.SecurityMode": "WPA-WPA2-Personal",
            "Settings.WIFI.Radio.1.Channel": "1",
            "Settings.WIFI.Radio.1.ChannelBandwidth": "20MHz",
            "Settings.WIFI.Radio.1.FrequencyBand": "2.4GHz",
            "Settings.WIFI.Radio.1.OperatingStandards": "b,g,n"
          },
          "subscriptionId": 7111
        }
    ]
}
```


NOTE: The JSON data structure above has been truncated for brevity. To view the entire data structure use the Device Manager RESTful Services tool (under UTILITIES) and enter the subscriber record GET URL to see a complete list of all services and settings returned in a subscriber JSON data structure.

Modify and Update the Subscriber Record

Modify the Settings in the JSON data structure returned above and then PUT the record back.

HTTP Command: PUT

URL: http://ACSName.smartrg.com/prime-home/api/v1/subscribers/3551

GET Device Actions

See the "Error! Reference source not found." section above.

Force Outbound Sync of New Settings

See the "Force outbound sync of new settings" above.

DELETE an existing subscriber/device record

Entries in the Device Manager ACS database commonly take the form of subscriber/device record pairs. Deleting record pairs must be done in device -> subscriber order. The following resource endpoints and JSON data structures shall be used to DELETE an existing subscriber/device record.

Query the Subscriber

See the "Query the Subscriber" section above.

DELETE the Device Record

Device records are identified by a unique handle known as "deviceId." The deviceId needed to reference a device is returned in the query above. See the highlighted deviceId.

HTTP Command: DELETE

URL: http://ACSName.smartrg.com/prime-home/api/v1/devices/6321

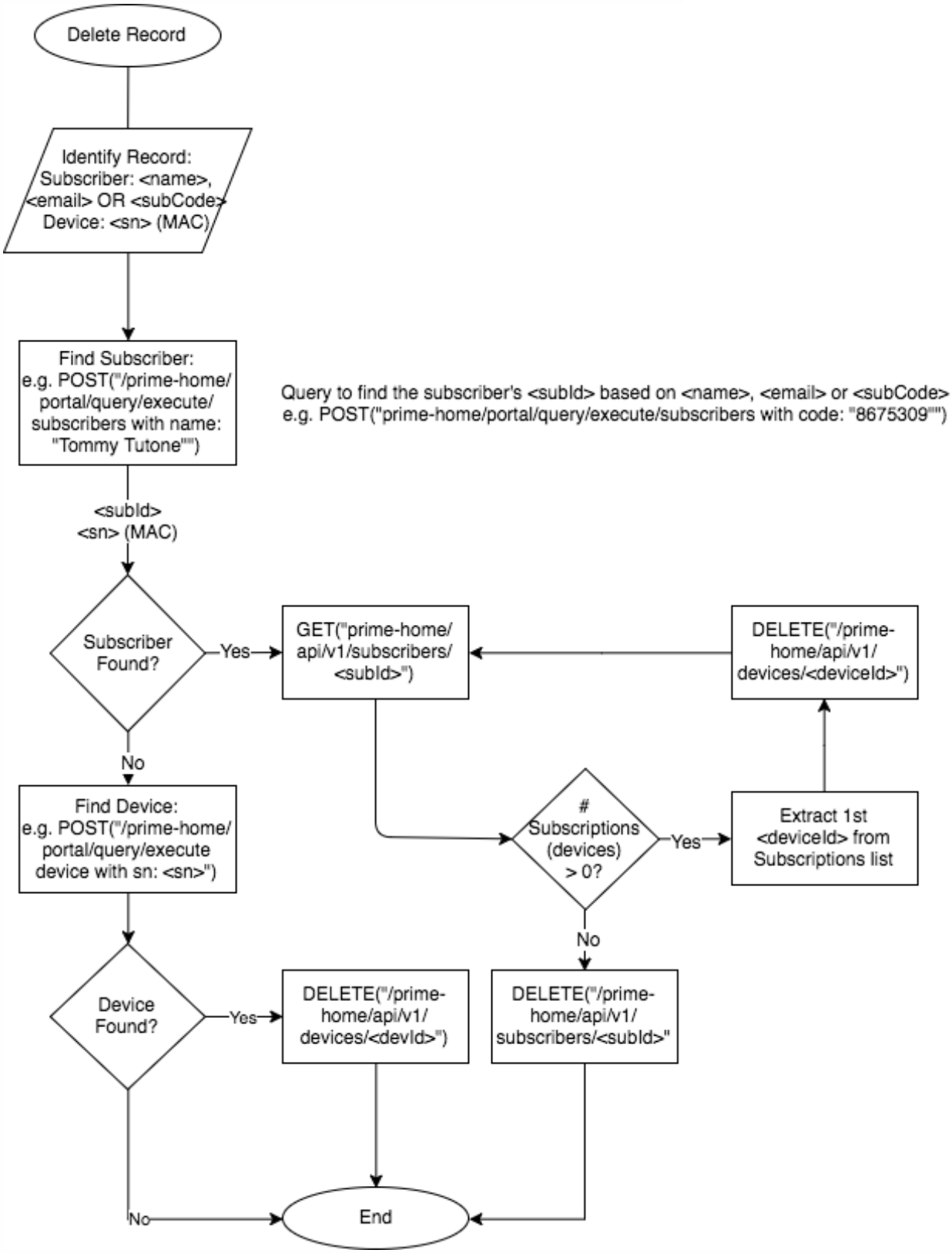
DELETE the Subscriber Record

Subscriber records are identified by a unique handle known as a "subscriberId." The subscriberId needed to reference a subscriber record is returned in the query above. See the highlighted subscriberId.

HTTP Command: DELETE

URL: http://ACSName.smartrg.com/prime-home/api/v1/subscribers/3551

DELETE Existing Subscriber/Device Record Logic Diagram



REVISION HISTORY

Version

VERSION	DATE	AUTHOR	DESCRIPTION
1.0	September 22, 2017	Patrick Maloney	Initial document
2.0	November 16, 2017	Patrick Maloney	• Added logic diagrams for the Create Subscriber/Device Record and DELETE Subscriber/Device Record processes • Truncated long API call responses for readability • Added notes showing how to use the Device Manager's REST tool • Converted to standard document template
2.0.1	January 16, 2018	Patrick Maloney	• Corrected version of the "NOTE: The JSON data structure above has been truncated for brevity. To view the entire data structure use the Device Manager RESTful Services tool (under UTILITIES) and enter the subscriber record GET URL to see a complete list of all services and settings returned in a subscriber JSON data structure. • Modify and Update the Subscriber Record" URL (v1 instead of v2) • Minor formatting modifications
2.1.0	February 9, 2018	Patrick Maloney	• Changed subscriber/device create order of operations • Corrected subscriber/device record parameter values • Modified create subscriber/device logic flow diagram to account for above changes • Added RESTful Service Tool section
2.1.1	December 13, 2018	David Busey	• Corrected Create Subscriber endpoint URL on pg 3 from /v2 to /v1
2.1.2	January 15, 2019	David Busey	• Complete overhaul of pgs 9-12. Sections named, "GET Device Actions" and "Force Outbound Sync of New Settings" replaced with new, more detailed instructions.