# Verizon Activation Process
## Activate Verizon SIMs using SKU (API)

Thomas Kurcz

Solutions Engineering Manager, IoT Connectivity

Telit

# Introduction

The purpose of this document is to show the reader how to activate Verizon SIMs on Telit's platform using Telit APIs and the Verizon SKU of the device.

The Verizon SKU is an identifier assigned by Verizon to a device when the device is certified by Verizon to operate on their network.

Please note that this document assumes that the reader already understands how to use API calls on Telit's platform.  More information about Telit's platform and APIs can be found at **https://docs.devicewise.com/Content/home.htm**

# Verizon Activation Process

The first step to activating Verizon SIMs is to verify that the SIMs you want to activate are in an "inventory" state. "Inventory" is the state that Verizon SIMs are set to when ordered from Telit.

The cdp.connection.find API can be used to determine the SIM state. Here is an example JSON request using the cdp.connection.find call for SIM ICCIDs 891480000388314XXX1 and 891480000388314XXX2:

```
{
        "1": {
                "command": "cdp.connection.find",
                "params": {
                        "iccid": "891480000388314XXX1"
}
},
        "2": {
                "command": "cdp.connection.find",
                "params": {
                        "iccid": "891480000388314XXX2"
}
}
}
```

# Verizon Activation Process

The result of the cdp.connection.find call shows everything we know about the SIMs including the SIM state.  Here is a portion of the result showing that the status is set to "inventory" for each SIM:

```
"1": {
    "success": true,
    "params": {
        "iccid": "8914800000388314XXX1",
        "carrier": "Verizon Wireless",
        "status": "inventory",
"2": {
    "success": true,
    "params": {
        "iccid": "8914800000388314XXX2",
        "carrier": "Verizon Wireless",
        "status": "inventory",
```

# Verizon Activation Process

The next step is to set the SKU*.  An API call to use for this purpose is cdp.connection.batch.update.  Here is an example of the JSON request that you would use to set a SKU of VZW080000XXXXX for SIM ICCIDs 891480000388314XXX1 and 891480000388314XXX2:

```
{
"1": {
"command": "cdp.connection.batch.update",
"params": {
"data":[
{
"cdp": {
"sku": "VZW080000XXXXX"
},
"iccid": "891480000388314XXX1"
},
{
"cdp": {
"sku": "VZW080000XXXXX"
},
"iccid": "891480000388314XXX2"
}
]
}
}
}
```

* The Verizon SKU is an identifier assigned by Verizon to a device when the device is certified to operate on their network.  Verizon device certifications are managed through Verizon's ODI portal (https://opendevelopment.verizonwireless.com/).

# Verizon Activation Process

The result of the cdp.connection.batch.update call provides a simple acknowledgment that the API was formatted correctly and accepted by our platform.  Here is what the result should look like:

```
{
    "1": {
        "success": true,
        "params": {
            "count": 2
        }
    }
}
```

# Verizon Activation Process

The next step is to verify that the SKUs were set correctly in our platform. The exact same cdp.connection.find API JSON that you used earlier to determine the SIM status can now be used to verify if the SKUs were set correctly. Here is the example JSON request for SIM ICCIDs 8914800000388314XXX1 and 8914800000388314XXX2:

```
{
        "1": {
                "command": "cdp.connection.find",
                "params": {
                        "iccid": "8914800000388314XXX1"
}
},
        "2": {
                "command": "cdp.connection.find",
                "params": {
                        "iccid": "8914800000388314XXX2"

}
}
}
```

# Verizon Activation Process

The result of the cdp.connection.find call shows everything we know about the SIMs.  This includes the SKU now that you set it.  Here is a portion of the result showing the VZW080000XXXXX SKU for each SIM:

```
"1": {
    "success": true,
    "params": {
        "iccid": "8914800000388314XXX1",
        "carrier": "Verizon Wireless",
        "status": "inventory",
        "sku": "VZW080000XXXXX"
"2": {
    "success": true,
    "params": {
        "iccid": "8914800000388314XXX2",
        "carrier": "Verizon Wireless",
        "status": "inventory",
        "sku": "VZW080000XXXXX"
```

# Verizon Activation Process

The next step is to activate the SIMs.  This is accomplished using the cdp.connection.batch.update call.

You can choose a status of "testing" or "activated".  Note that no matter what option you choose, the state will go into a "testing" state once the SIMs are activated.  The "testing" state is an active state that will automatically transition to an "activated" (billing) state after 150 days, 60KBs of data consumption or 15 SMS.  For the purposes of this doc, the state of "testing" will be used when activating SIMs.

# Verizon Activation Process

Here is an example of the JSON request you would use to change the status of SIM ICCIDs 891480000388314XXX1 and 891480000388314XXX2 to a "testing" state:

```
{
"1": {
"command": "cdp.connection.batch.update",
"params": {
"data":[
{
"iccid": "891480000388314XXX1",
"status": "testing"
},
{
"iccid": "891480000388314XXX2",
"status": "testing"
}
]
}
}
}
```

Here is an example of the JSON response you would see after changing the status to a "testing" state:

```
{
    "1": {
        "success": true,
        "params": {
            "count": 2
        }
    }
}
```

# Verizon Activation Process

The SIM activations typically takes a couple of minutes.  Before you can start using the SIMs, you need to make sure that they are in a "testing" state.  To check and see the status, you can use cdp.connection.find call.

Here is an example of the JSON request that you would use to see if the 891480000388314XXX1 and 891480000388314XXX2 ICCIDs are in a "testing" state:

```
{
        "1": {
                        "command": "cdp.connection.find",
                        "params": {
                                        "iccid": "891480000388314XXX1"
}
},
        "2": {
                        "command": "cdp.connection.find",
                        "params": {
                                        "iccid": "891480000388314XXX2"

}
}
}
```

# Verizon Activation Process

If you send the cdp.connection.find call to check the status right after you sent the cdp.connection.batch.update call to activate, there is a good chance that you will see the status of the SIMs show a state of "activating".  "activating" is a transitional state that you will see when the SIMs are in the process of being activated.

Here is portion of the JSON response to the cdp.connection.find call that you just ran showing a status of "activating" for the two SIMs:

```
"1": {
      "success": true,
      "params": {
          "iccid": "8914800000388314XXX1",
          "carrier": "Verizon Wireless",
          "status": "activating",
          "sku": "VZW080000XXXXX"
    "2": {
          "success": true,
          "params": {
              "iccid": "8914800000388314XXX2",
              "carrier": "Verizon Wireless",
              "status": "activating",
              "sku": "VZW080000XXXXX"
```

# Verizon Activation Process

Again, what you are looking for is a status of "testing". Once the SIMs have a "testing" status, you know that they have been successfully activated.

Here is portion of the JSON response to the cdp.connection.find call showing a status of "testing" for the two SIMs:

```
"1": {
        "success": true,
        "params": {
            "iccid": "8914800000388314XXX1",
            "carrier": "Verizon Wireless",
            "status": "testing",
            "sku": "VZW080000XXXXX"
    "2": {
        "success": true,
        "params": {
            "iccid": "8914800000388314XXX2",
            "carrier": "Verizon Wireless",
            "status": "testing",
            "sku": "VZW080000XXXXX"
```

# Verizon Activation Process (multiple SIMs/devices)

You now have active SIM(s) that can be installed in Verizon devices that have the same SKU that you set earlier in this procedure.  If your application only requires Internet Direct (device provided with access to the Internet), you can begin device setup as defined by your device manufacturer and start using the device.

If you are a Telit VPN customer, you can also begin device setup as defined by your device manufacturer, but before you can communicate with the device through your VPN, you will first need to ask Telit to associate the SIM(s) you just activated with your VPN.  To get this done, send the ICCID(s) that you just activated to support-iotconnectivity@telit.com.  Doing so will open a case that will be routed to our engineering department.  In the body of the email, be sure to include the ICCID(s) and a note asking engineering to associate the SIM(s) with your VPN.  Once engineering completes the work, they will send you an email containing the private IP address(es) that you will need to use to access the device(es).

# Thank You!

Any feedback/questions/comments please email support-iotconnectivity@telit.com