Oracle® Cloud

Known Issues for Oracle Internet of Things Cloud Service

Release 23.3.1

E74898-51

July 2023

Known Issues

Topics

Oracle IoT Cloud Service Management Console

Non-Admin Application Users Need Explicit Viewer Privilege

IOT-93795: If a non-admin user logs in to the management console (`/ui`) without explicit **Viewer** role, the following error message appears: "The user is forbidden from performing this action. Please make sure the user is authorized to perform the action".

Workaround: Assign the **Viewer** role to those non-admin users of your IoT application, who need to use the management console (`/ui`).

Creation of Explorations is Deprecated

IOT - 83159: The stream exploration feature in Oracle IoT Cloud Service release 20.4.3 or later is deprecated and will be removed in a future release.

Workaround: None

Export and Import of Applications Is No Longer Supported

IOT-83695: Exporting or importing an IoT Application from the Management Console of Oracle IoT Cloud Service release 21.2.1 or later can no longer be performed. This feature has been removed.

Workaround: Use the export and import options at the organization level within an IoT Application.

URI and Datetime Types Not Allowed in the Device Policy Editor

IOT-76315: When adding a formula for a function in a policy pipeline of a device policy, the formula validator does not allow URI and datetime types.

Workaround: None.

No Data Displayed in the Analyzed Messages Chart

IOT-59394: In Oracle Internet of Things Cloud Service release 19.1.1, the analyzed messages chart on the application dashboard does not show any data despite the data being available. .

Workaround: If analyzed messages data is important to your business, contact Oracle support for assistance getting the data to display.

Batch Registration Errors Occur When Using CSV Files Edited in Microsoft Excel

IOT-47757: Uploading Microsoft Excel edited files for batch registration may result in errors when using the following process:

1. Download the `.csv` file template from the Oracle IoT Cloud Service server.

2. Edit the `.csv` file in Microsoft Excel.

3. Upload the `.csv` file to the Oracle IoT Cloud Service server for batch registration.

Workaround: Use a plain text editor, such as the Notepad application, for editing the .csv template.

Device Software Gets Deleted From Device When Uploaded Software Is Deleted

IOT-14755: Using the **Software** page in the **Devices** section of the Oracle IoT Cloud Service Management Console, if you upload a device software to the Oracle IoT Cloud Service software repository, install it on a gateway device, and then delete the uploaded software, the device software appears to have been deleted from the gateway device. In fact, the device software is still installed in the gateway device, but

the **Software** tab in the gateway device's details page no longer displays the information about the installed software.

Workaround: None.

Oracle IoT Cloud Service Client Software Libraries

C POSIX Samples That Use the Messaging APIs Do Not Build with the Provided Binary Bundle

IOT-19800: The `directly_connected_device_sample.c` and `gateway_device_sample.c` source files that are located in the `<client-lib-install-dir>/posix/samples/make/src/advanced` folder use the Oracle IoT Cloud Service Client Library Messaging API libraries only. These source files can't be built using the default binary bundle that is provided with the C POSIX Client Library download because that binary bundle contains libraries that are only supported by the Oracle IoT Cloud Service Client Library Device Virtualization API.

Workaround: Build the Oracle IoT Cloud Service Client Library source bundle for the C POSIX platform using the Messaging API libraries. Use the following steps to guide you:

1. Navigate to the `<client-lib-install-dir>/posix/make/` folder.

2. Enter the following at the command line:

   ```
   make clean all CONFIGURATION=posix LOW_LEVEL_THREAD_SAFETY=true
   GATEWAY=false MESSAGE_DISPATCHER=true VIRTUALIZATION_SUPPORT=false
   ```

3. Use the newly built source bundle to build the `directly_connected_device_sample.c` and `gateway_device_sample.c` source files that are located in the `<client-lib-install-dir>/posix/samples/make/src/advanced` folder.

Network Provisioning Multicast Discovery Fails on Some Android Devices

IOT-39166: When using network provisioning from an android application to automatically discover registered devices, the UDP multicast discovery request is currently failing on certain devices.

The discovery messages are currently failing on the emulator, Samsung Galaxy S7, Nexus 4, Nexus 9, and Samsung Note 10.1.

The discovery messages are currently working for Samsung Galaxy S6, Alcatel A30, HTC Desire 625, Samsung Galaxy S4, and Nexus 6.

Workaround: None.

Java SE Client Library OSGi Sample Fails to Execute

IOT-40999: The OSGi sample compiles and installs correctly, but returns the following runtime error:

```
sun/net/www/protocol/https/Handler.
Caused by: java.lang.ClassNotFoundException:
sun.net.www.protocol.https.Handler not found by
oracle.iot.client.device
```

Workaround: Use the following steps as the workaround:

1. Download and unzip the Oracle IoT Cloud Service Java SE Client Software Library source distribution.

    You can download the source distribution from the Oracle Internet of Things Cloud Service Client Software Libraries download page.

2. Change to the `iotcs/csl/javase` directory.

3. Edit the `build.gradle` file.

    Change the following line:

    ```
    def OSGI_IMPORT_PACKAGE =
    "javax.crypto,javax.crypto.spec,javax.net.ssl,org.json"
    ```

    to the following:

    ```
    def OSGI_IMPORT_PACKAGE =
    "javax.crypto,javax.crypto.spec,javax.net.ssl,org.json,sun.net.www.p
    rotocol.https"
    ```

4. Run `gradle` to rebuild the library jar files.

5. Include –
    `Dorg.osgi.framework.system.packages.extra=sun.net.www.protocol.https` on the java command line when running the OSGi framework.

    For example:

    ```
    java -
    Dorg.osgi.framework.system.packages.extra=sun.net.www.protocol.https
     -Dsample.name=GatewayDeviceSample -Dsample.args="SAMPLE-
    PROVISIONING-FILE Passphrase123" -jar
    /home/myaccount/felix/bin/felix.jar
    ```

Oracle IoT Cloud Service Gateway

Gateways Connected as Indirectly Connected Devices Cause a Stack Overflow

IOT-58661: When a gateway is connected to Oracle Internet of Things Cloud Service as an indirectly connected device, a stack overflow occurs. This is due the gateway assuming the roles of both the parent and child device.

Workaround: Do not connect gateways to Oracle Internet of Things Cloud Service as indirectly connected devices. In future releases, this issue will be automatically detected and prevented.

Misleading Error Message When Installing Already Installed Software

IOT-18114: If you try to remotely install software on a gateway where there already exists software with that same version, the error that you get is not clear. Any failure in the installation raises the same error message, regardless of the cause.

Workaround: Before installing software remotely on a gateway, verify if a version of that software is already installed using REST APIs.

Oracle IoT Cloud Service Connectors

Connectors Unable to Parse JSON Payloads Containing Attribute Names With a Period

IOT-97545: Connectors in the Oracle IoT Cloud Service Management Console are unable to correctly parse JSON payloads with attribute names containing a period (.), for example `position.speed`, `position.latitude`, and so on. The value of any JSON field containing a period (.) is always being read as `1`.

Workaround: None.

Messages for an MQTT Client Connector are Not Evenly Distributed Across All its Instances

IOT-64856: While creating an MQTT Client Connector, you can select **Scale Factor** as the number of connector instances to be launched. However, the messages received by the connector instances may not be evenly distributed. A single instance may receive most of the messages whereas the other instances may receive 1 or none.

Workaround: The distribution of messages to the individual connector instances is a function of the MQTT broker and is not determined by the connector. Some brokers evenly distribute messages across each connector instance, while some duplicate messages to each instance. If your MQTT broker cannot distribute messages evenly across the instances, then use a single scale instance of the MQTT Client connector. First, test the distribution of messages by the MQTT broker, and if the broker cannot evenly distribute the messages across the connector instances then, select **Scale Factor** as **1**.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.