Tech Note

Ayla File Properties

Version: 1.0

Date Released: September 18, 2020

Ayla Networks

Referenced Documents

Ayla Networks does not supply all documents that are referenced in this document with the equipment. Ayla Networks reserves the right to decide which documents are supplied with products and services.

Contact Information

Ayla Networks TECHNICAL SUPPORT and SALES

Contact Technical Support: https://support.aylanetworks.com
or via email at support@aylanetworks.com

Contact Sales: https://www.aylanetworks.com/company/contact-us

Ayla Networks REGIONAL OFFICES

For a Complete Contact List of Our Offices in the US, China, Europe, Taiwan, and Japan:
https://www.aylanetworks.com/company/contact-us

Table of Contents

1 Ayla File Properties

File properties are used to send large amounts of data in files that are stored in Amazon S3 (Simple Storage Service) for customers to parse, interpret, or use on their devices. Some customers, for example, wish to send files with specific images and/or information to use on their devices, and other customers are continuously collecting large amounts of specific device data in files for analysis. File properties enable these customers to send fewer data transactions, especially when continuously collecting and sending data, which reduces transaction costs.

In comparison to Ayla's message properties, which can also be used when sending smaller files (512 KB or smaller), file properties should be used for files containing large binary images and data (binary large objects) and other compressed data formats. Whereas, message properties should be used for string, JSON, and smaller binary data formats.

Unlike the datapoints for Ayla's other device property base types (boolean, integer, decimal, string, and message), we do not keep a history of file property datapoints once the file is downloaded from Amazon S3. It is therefore important to understand the dataflow operations for file properties to ensure that application users or devices are marking the file property datapoint as complete and fetched so that new files can be uploaded and downloaded.

This Tech Note describes file properties and explains the file property dataflow operations.

2 File Property Characteristics

Size:

File properties can be a maximum of 4 GB long. For comparison, the maximum size of message property datapoints is 512 KB and of string property values is 1024 bytes.

Storage:

Files sent using the file property are stored in Amazon S3 (Simple Storage Service) so that customers can download the files. S3 storage allows objects as large as 5 GB to be uploaded with a PUT command. A file can only be downloaded until its file property datapoint is marked as fetched. Once it is marked as fetched, the file property datapoint is deleted so that new files can be uploaded and downloaded.

Limitations:

- When implementing file properties in a host application on a device, file uploads and downloads are longer processes relative to other device (microcontroller unit) operations. During the upload and download processes, no other property updates can be sent to the device. Unless, the Ayla module notifies the microcontroller unit (MCU) of pending property updates with the AD_PROP_NOTIFY opcode, which is used if you want the MCU to abort the file transfer to receive the other pending updates. Refer to the *Ayla Module and MCU Interface Guide* for more information on file datapoint operations relative to the Ayla Embedded Agent.

- Batch updates do not support file properties.
- File property datapoints can only be read until they are marked as “fetched,” after which they are deleted.

Metadata

- File properties support metadata. For more information on adding metadata to the file property, refer to the *Ayla Module and MCU Interface Guide, Section 7.2.2.*
- You can add a maximum of 4 key-value pairs of metadata. The Design Kit MCU Demo Source Package on Ayla Connection provides demo code and the file prop_dp.c to show you how to add metadata to a from-device property. Click here to obtain this package.
- For a to-device file property, the Ayla Mobile Software Development Kit (SDK) supports adding a map with key-value pairs when sending the file property datapoint. If you are interested in this SDK, let your Ayla Customer Success Representative know.

3 How File Properties Work in the Ayla Platform

Following are four main actions in the Ayla Platform when implementing file properties:

1. Create the file property datapoint using the Ayla REST APIs or Ayla Developer Portal.
2. Upload the file to Amazon S3.
3. Mark the file property datapoint as complete so that it is available to download the file.
4. After downloading the file (and when you are ready to upload/download your next file), mark the file property datapoint as fetched so that it is deleted to allow for subsequent file uploads/downloads.

Based on these actions, this section describes:

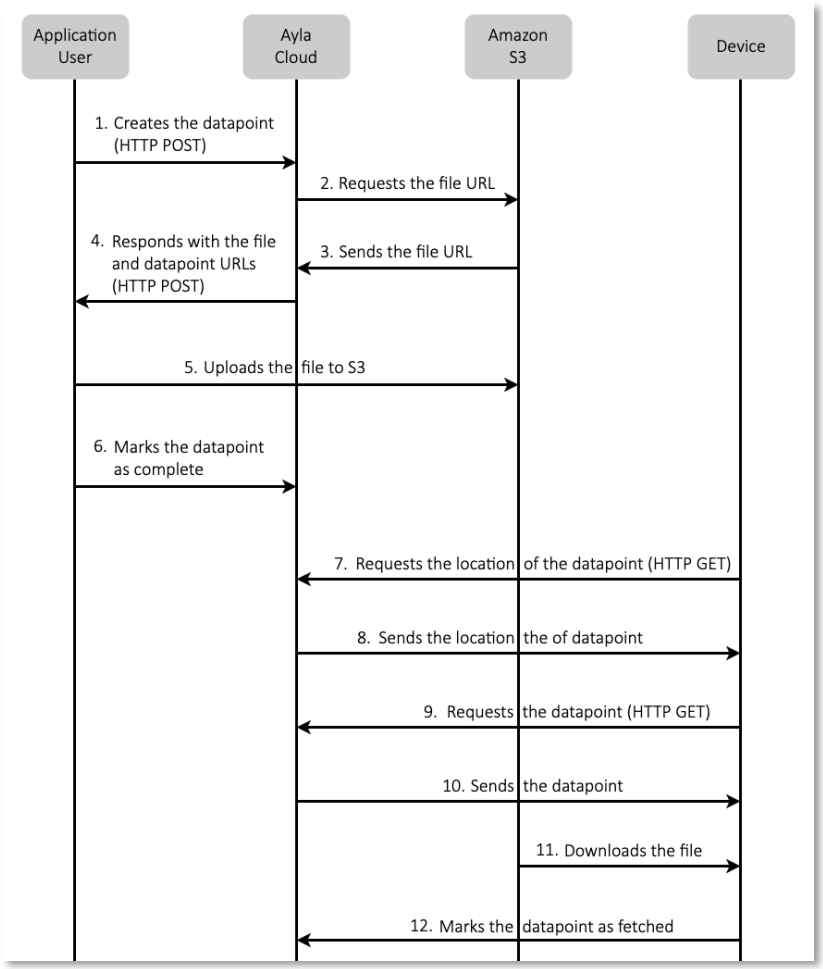
- How file property datapoints are used to upload and download files.
- The steps to create file properties and their datapoints.

3.1 File Property Dataflows

The diagrams in this section show how file properties are implemented to upload and download files in the Ayla Platform.

3.1.1 To-Device Property Dataflow

The following dataflow diagram shows a to-device configuration for the file property:

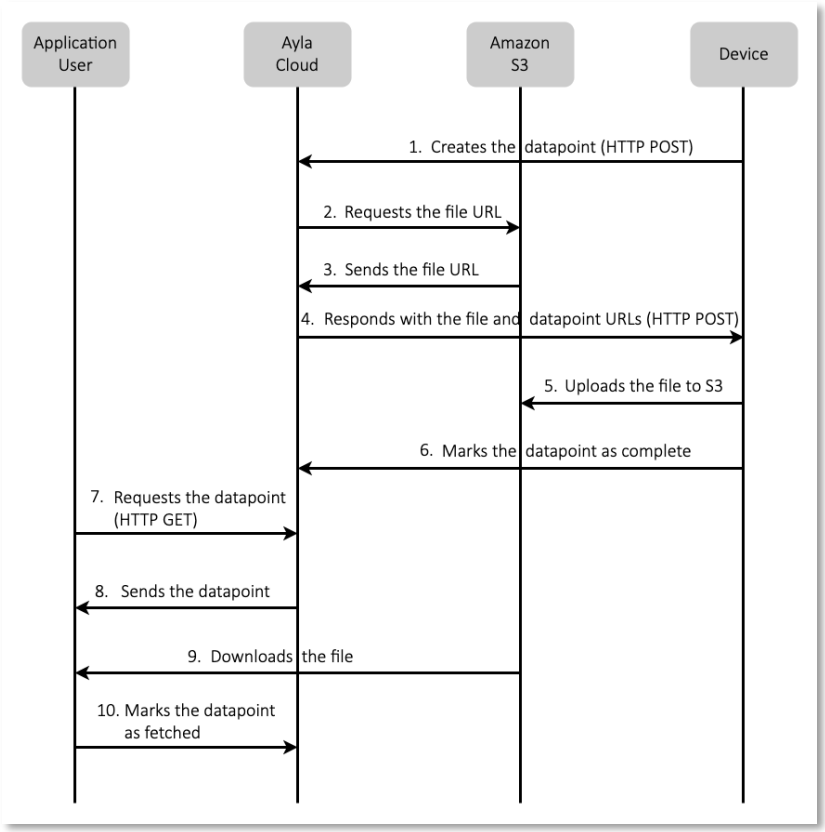


1. The application user creates the file property datapoint to upload the file. (The value for the file property datapoint is a location URL for the Ayla Cloud.) This is done using the HTTP POST REST API.
2. The Ayla Cloud requests the file URL from the Amazon S3 bucket. (The file URL is the location where the file will be stored in S3.)
3. Amazon S3 responds to the Ayla Cloud with the file URL.
4. The Ayla Cloud sends a POST response with the file datapoint URL and the file URL to the application user.
5. The application user uploads the file (via Ayla REST APIs) to the Amazon S3 bucket using the file URL.
6. At this point, the application user must mark the file property datapoint complete so that the device can retrieve (fetch) this datapoint to download the file. Marking the file property datapoint complete also indicates that the file was successfully uploaded to Amazon S3.
7. The device requests (via HTTP GET) the file property datapoint from the Ayla Cloud.
8. The Ayla Cloud sends the file property datapoint to the device. (The location URL is the file property datapoint value).

9. Using the location URL, the device requests (via HTTP GET) the file property datapoint from the Ayla Cloud.
10. The Ayla Cloud sends the file property datapoint to the device.
11. The file is downloaded from the Amazon S3 bucket to the device.
12. Once the file is downloaded to the device, the device must mark the file property datapoint as fetched. Marking the datapoint as fetched is mandatory because it deletes the datapoint from the queue and enables the device to fetch the next file property datapoint for the new file (the first in, first out method).

3.1.2 From-Device Property Dataflow

The following dataflow diagram shows a from-device configuration for the file property



1. The device creates a file property datapoint (via HTTP POST) in the Ayla Cloud.
2. The Ayla Cloud requests the file URL from the Amazon S3 bucket. (The file URL is the location where the file will be stored in S3.)
3. Amazon S3 responds to the Ayla Cloud with the file URL.
4. The Ayla Cloud sends a POST response with the file location URL to the device.
5. The device uploads the file to the Amazon S3 bucket using the file URL.

6. At this point, the device must mark the file property datapoint complete so that the application user can retrieve (fetch) this datapoint to download the file. Marking the file property datapoint as complete also indicates that the file was successfully uploaded to Amazon S3.
7. The application user requests (via HTTP GET) the file property datapoint from the Ayla Cloud.
8. The Ayla Cloud sends the file property datapoint to the application user. The location URL is the file property datapoint value.
9. The application user downloads the file from the Amazon S3 bucket using the file location URL.
10. Once the file is successfully downloaded, the application user marks the file property datapoint as fetched if the user plans to upload/download a new file. Marking the datapoint as fetched is mandatory because it deletes the datapoint from the queue and enables the application user to fetch the next file property datapoint for the new file (the first in, first out method).

NOTE In the Ayla Mobile Software Development Kit (SDK), there are two APIs for fetching a file property. This provides flexibility to the developers. They can provide an application that always marks a file fetched after it is retrieved, or they can just fetch he file and provide marking the file datapoint as fetched (deleting) as a separate screen or clean-up function. If you are interested in the Ayla Mobile SDK, let your Ayla Customer Success Representative know.

3.2 Creating and Managing File Properties

- Use the REST APIs.
- Use the Ayla Developer Portal.

3.2.1 Using Rest APIs

Creating a File Property Datapoint (To-Device Configuration)

The application user completes the following steps to create a file property datapoint (in a to-device configuration):

1. Create the file property datapoint using the POST request:

```
POST   apiv1/dsns/:dsn/properties/:property_name/datapoints
```

JSON Example:

```
curl -X POST -H"Authorization: auth_token 0af85be8725c4e50ab1244e2233b0f80" -d
""-H "Content-Type: application/xml" \

https://ads.aylanetworks.com/<hostname>/apiv1/dsns/AC000W123456789/properties/10
/datapoints.json
```

NOTE: The **host name** (in the curl example above) is dynamic and specific to the environment and region, for example:

- US Development: https://ads-dev.aylanetworks.com
- US Field: https://ads-field.aylanetworks.com
- EU Field: https://ads-field-eu.aylanetworks.com

JSON Response:


```
{

    "datapoint": {

        "updated_at": "2015-01-30T20:49:56Z",

        "created_at": "2015-01-30T20:49:56Z",

        "echo": false,

        "closed": false,

        "file": "https://s3.amazonaws.com/ayla-device-development-
oem_id/datapoint-8c7c0510-a8c1-11e4-b494-
29a01a2bf202?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1422654597&Signature=Dh
1%2F6t5lNlCEdDQOmOizcrULZr8%3D",

        "value": "https://ads.aylanetworks.com
/apiv1/devices/5/properties/p1/datapoints/8c7c0510-a8c1-11e4-b494-
29a01a2bf202.json"

    }

}
```


2. Upload the file using the PUT request:

```
curl -X PUT -v  -T  /home/Documents/sample-file.pdf -H "Content-Type:
application/octet-stream" "https://s3.amazonaws.com/ayla-device-development-
oem_id/datapoint-8c7c0510-a8c1-11e4-b494-
29a01a2bf202?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1422654597&Signature=D
h1%2F6t5lNlCEdDQOmOizcrULZr8%3D"
```

NOTE: Notice the “Expires” query string in the above curl example. This is how long the file datapoint URL is valid before it expires. If it expires, your upload request will receive a 403 error. Use a GET request to obtain the file datapoint again, and the request should return a new <file>, which is the location URL (file property datapoint). This <file> section provides a new expiration time and signature. Refer to the following example:

JSON Example:

```
curl -v -X GET -H"Authorization: auth_token 0af85be8725c4e50ab1244e2233b0f80" -
H "Content-Type:
application/xml" http://localhost:3000/apiv1/devices/5/properties/p1/datapoints
/06437fd0-a8c8-11e4-b54e-29a01a2bf202.json
```

JSON Response:


```
{

    "datapoint": {

        "updated_at": "2015-01-30T21:36:17Z",

        "created_at": "2015-01-30T21:36:17Z",
```

```
        "echo": false,

        "closed": false,

        "value":
"http://deviceservice.dev/apiv1/devices/5/properties/p1/datapoints/06437fd0-
a8c8-11e4-b54e-29a01a2bf202.json",

        "file": "https://s3.amazonaws.com/ayla-device-development-
oem_id/datapoint-06437fd0-a8c8-11e4-b54e-
29a01a2bf202?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1422658179&Signature=E
5UBD89SpZ4RjajfMeHR%2BxCnIUc%3D"

    }

}
```


3. After successfully uploading the file, close the datapoint in the cloud using a PUT request. This marks the action for this file property datapoint as complete and ready to be retrieved (fetched) by the device(s). After issuing this PUT request, the file property datapoint appears in the Ayla Developer Portal and Customer Dashboard.

   JSON Example:

   ```
   curl -X PUT -H"Authorization: auth_token 0af85be8725c4e50ab1244e2233b0f80" -d
   "<datapoint><closed>1</closed></datapoint>" -H "Content-Type:
   application/xml" http://localhost:3000/apiv1/devices/5/properties/p1/datapoints
   /be5efcb0-a8c4-11e4-b494-29a01a2bf202.json
   ```

   JSON Response:

   Returns a 200 upon successfully closing the file property data point.

Downloading the File and Marking the Datapoint as Fetched (From-Device *Configuration)*

When the device creates the file property (in a from-device configuration), the application user issues the following REST APIs to download the file and mark the file property datapoint as fetched:

- After the device marks the file property datapoint as complete, the application user issues a GET request to retrieve the file property datapoint and then download the file:

  JSON Example:

  ```
  curl -k -X GET -H "Authorization: auth_token <auth_token>" https://<device
  service url>/apiv1/devices/<device-id>/properties/file_down/datapoints/1063224
  ```


  ```
  {"datapoint":{"updated_at":"2015-01-21T16:40:53Z","created_at":"2015-01-
  21T16:40:53Z","echo":false,"closed":true,"value":"https://<device service
  url>apiv1/devices/<device-id>/properties/test/datapoints/43c561e0-a18c-11e4-
  8c86-b935dba94266.json","file":"https://<device service url>/datapoint-
  43c561e0-a18c-11e4-8c86-
  b935dba94266?AWSAccessKeyId=AKIAJAR3VCJIAFO6AXXX&Expires=1421863148&Signature=d
  rHshNV%2F2CVXbbhJenyONAp89dE%3D"}}
  ```


  Response Status Codes:

  200 – Successfully downloaded

  406 – Failed to access the datapoint

- Once the device retrieves this file property datapoint and downloads the file (using GET requests), the file property datapoint must be marked as fetched. This is because file property datapoints can only be fetched in a certain order for the devices. The order is to fetch the first closed and unfetched datapoint in the queue. Therefore, if the datapoint is not marked as fetched, it is not removed from the queue of unfetched file property datapoints and can erroneously be retrieved again.

  The PUT request is used to mark file property datapoint as fetched:

  JSON Example:

  ```
  curl -X PUT -H"Authorization: auth_token af85be8725c4e50ab1244e2233b0f80" -d
  "<datapoint><fetched>1</fetched></datapoint>" -H "Content-Type:
  application/xml" http://localhost:3000/apiv1/devices/5/properties/p1/datapoints
  /06437fd0-a8c8-11e4-b54e-29a01a2bf202.json
  ```

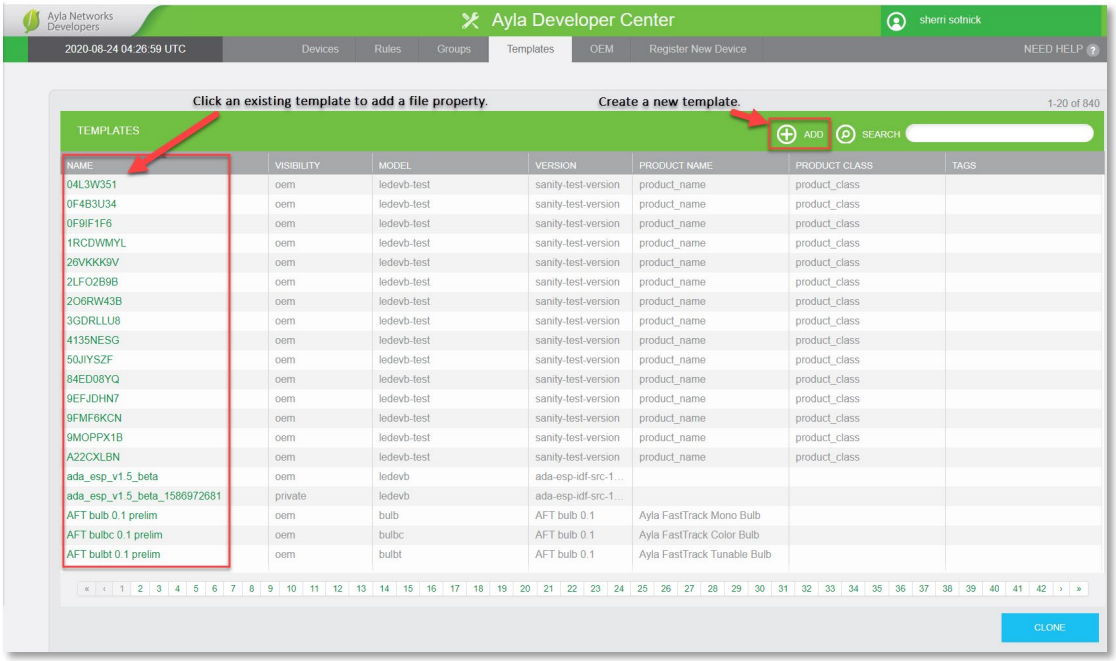
  JSON Response:

  Returns a 200 upon successfully fetching the file property data point.

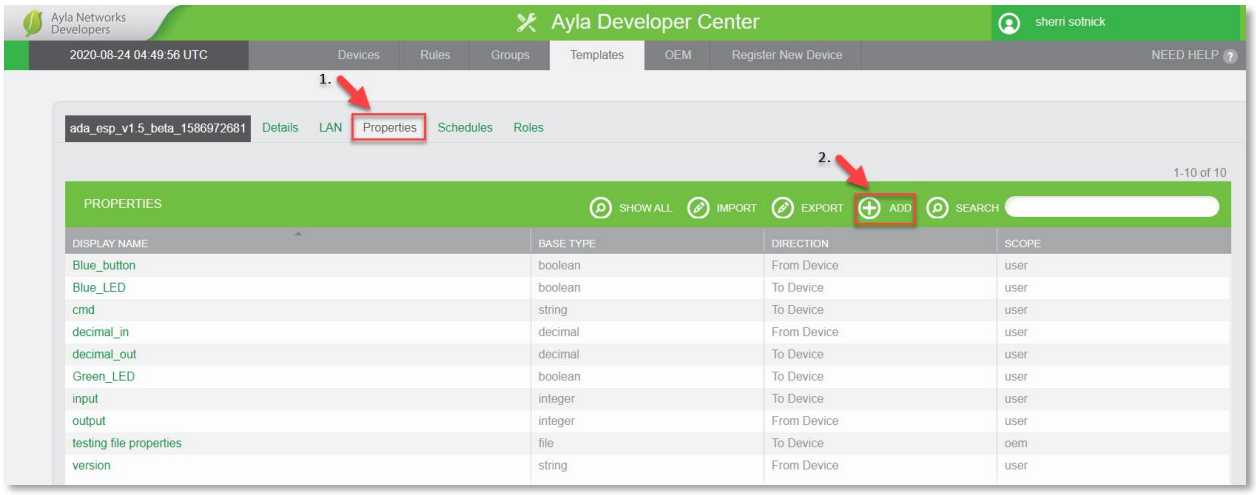
3.2.2 Using the Ayla Developer Portal to Create File Properties

You can create file properties using the Ayla Developer Portal and then manage them in Customer Dashboard as follows:

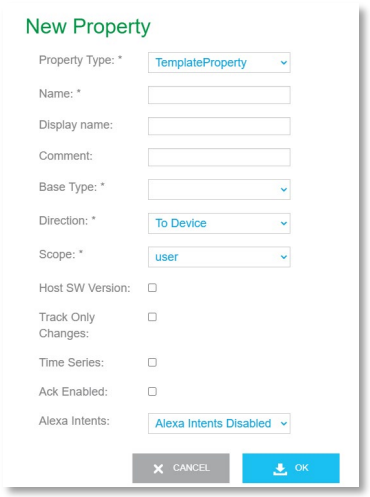
1. Navigate and sign in to the Ayla Developer Portal.
2. Click the **View My Devices** icon.
3. Click the **Templates** tab, and then click the name of template for which you wish to create a file property. If you wish to create a new template, click the **ADD** button, and once you have added your new template to the list, click its name to create a file property. (Refer to the following example.) Note that you can only add properties to templates with Private visibility.



4. Click the **Properties** tab, and then click the **ADD** button, as shown in the following example:



5. In the New Property dialog box (shown below), complete the mandatory fields at a minimum (as described below):



   a. For **Property Type**, select **Template Property**.

   b. For **Name**, enter a name that matches the property name of the host application on the device.

   c. For **Base Type**, select **file**.

   d. For **Direction**, select the appropriate option from the drop-down list:

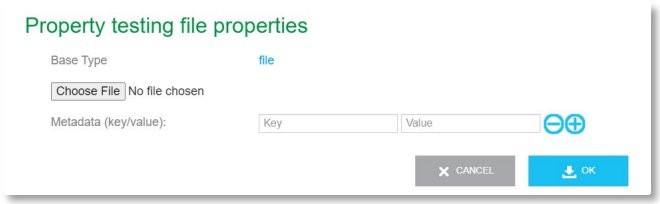
      - **To Device** (information flows to the device)
      - **From Device** (information flows from the device)

   e. For **Scope**, select an option from the drop-down list:

      - **user** (all users can see the property)
      - **oem** (only customers defined as the OEM in the portal can see the property)

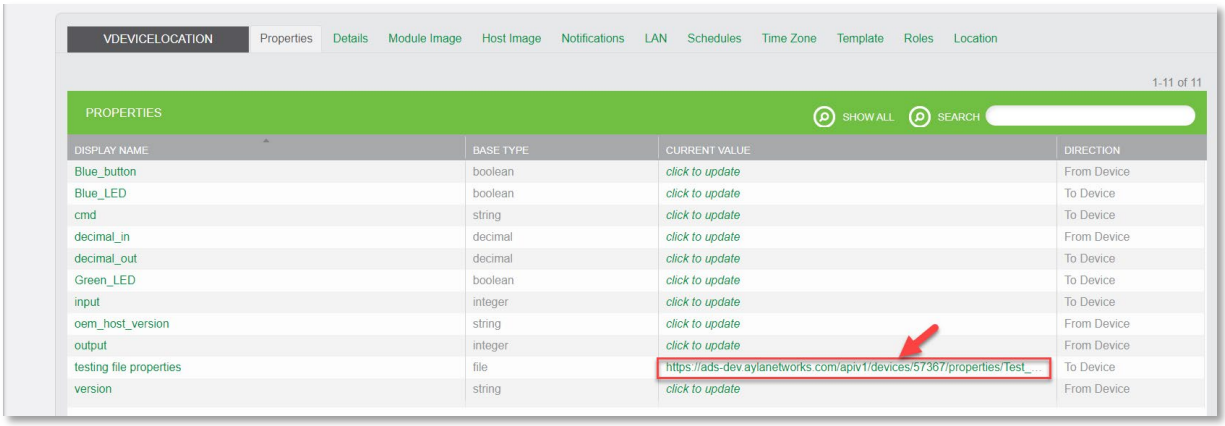
   f. Click **OK**.

6. Click the **Devices** tab, and then click the device to which you are either uploading or downloading the file. Use the search tool if needed to find the device.
7. Click the **Template** sub-tab (on the Devices tab) to associate or re-associate the template for which you created the new file property.
8. Click the **Properties** tab, and then find your new file property in the list.
9. For **Value**, click the **click to update** link.



10. Click **Choose File** to navigate to and upload the file.
11. Enter metadata if you wish, and then click **OK**.

    Notice in the following example that the file property datapoint displays as the Current Value for your file property. The value for the file property datapoint is a location URL representing the datapoint in the Ayla Cloud.

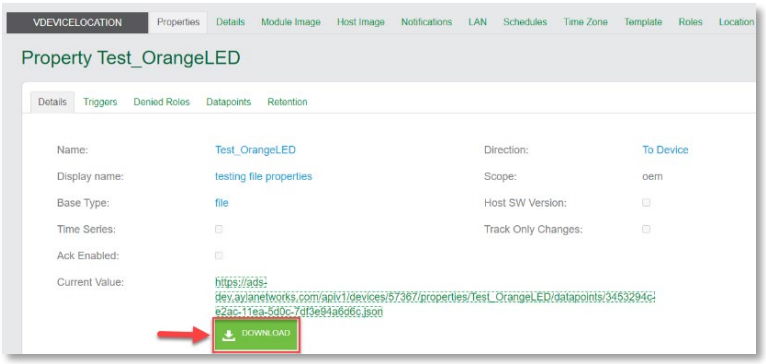


    Once the file is uploaded and you receive the file property datapoint (location URL), as shown in the example above, the Developer Portal marks the datapoint as complete, so that it is available for the device to fetch when downloading the file. This is equivalent to step 6 in the to-device dataflow diagram/explanation in the File Property Dataflow section.

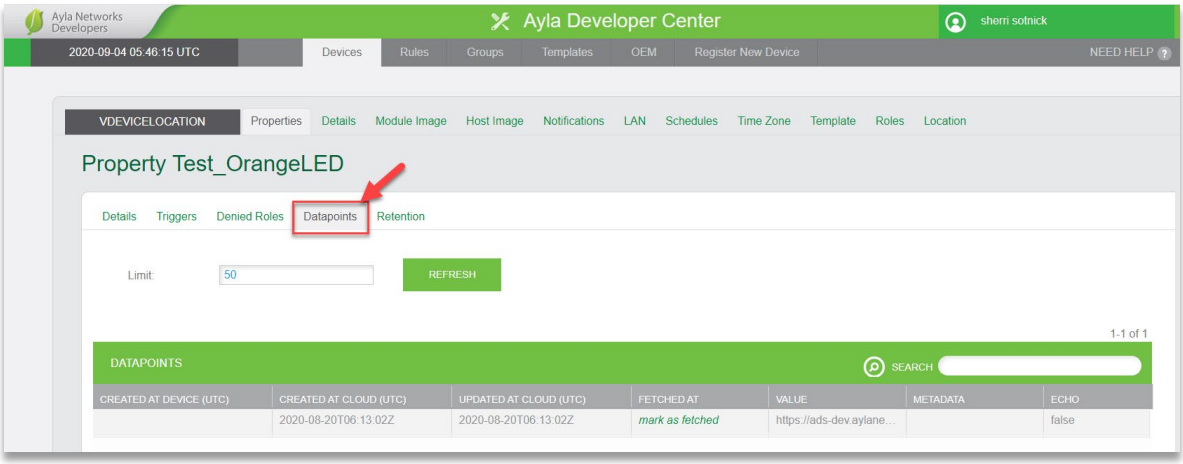
    If the device is in the US development environment and registered to your account, you can download the file locally via the Developer Portal as follows:

    a. Click the file property for the file you wish to download. (This is the Display Name column in the example above.)

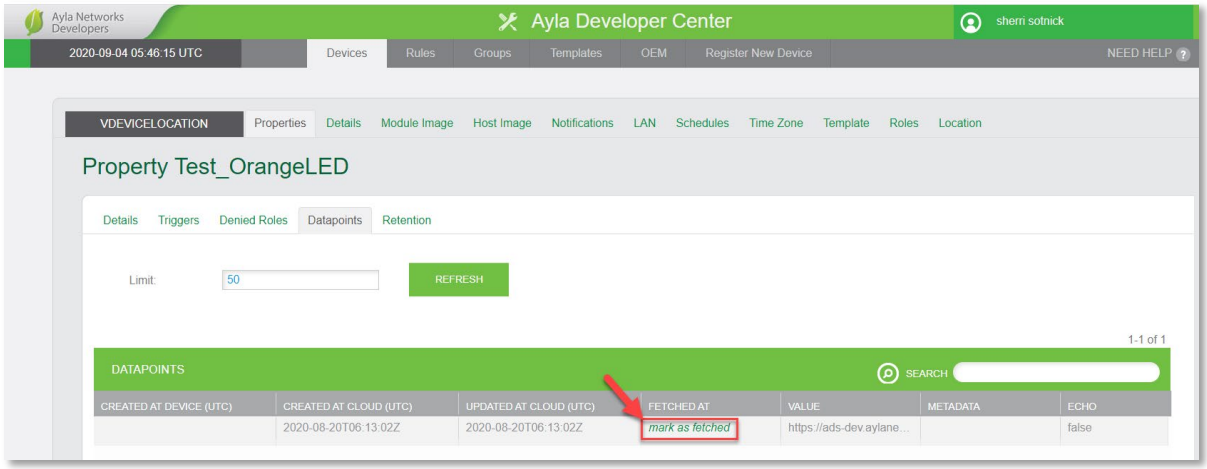
    b. Click the **DOWNLOAD** button, as shown in the following example:



12. Click your new file property (the Display Name) to open the Details tab for this property.
13. Click the **Datapoints** tab, as shown below:



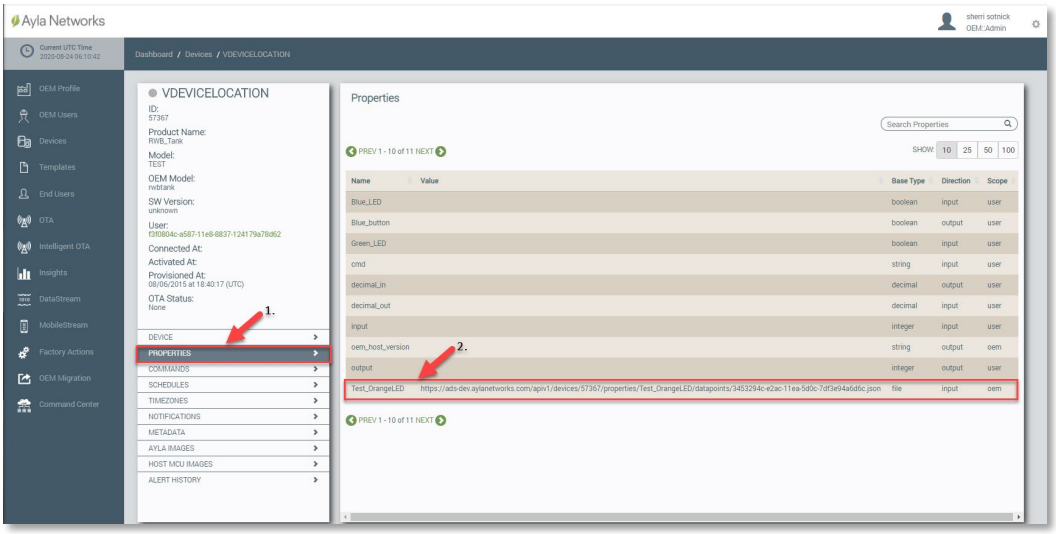
14. Click the **mark as fetched** link in the Fetched At column (shown below). You must mark the file property datapoint as fetched so it is deleted, enabling you to upload/download new files via a new file property datapoint.



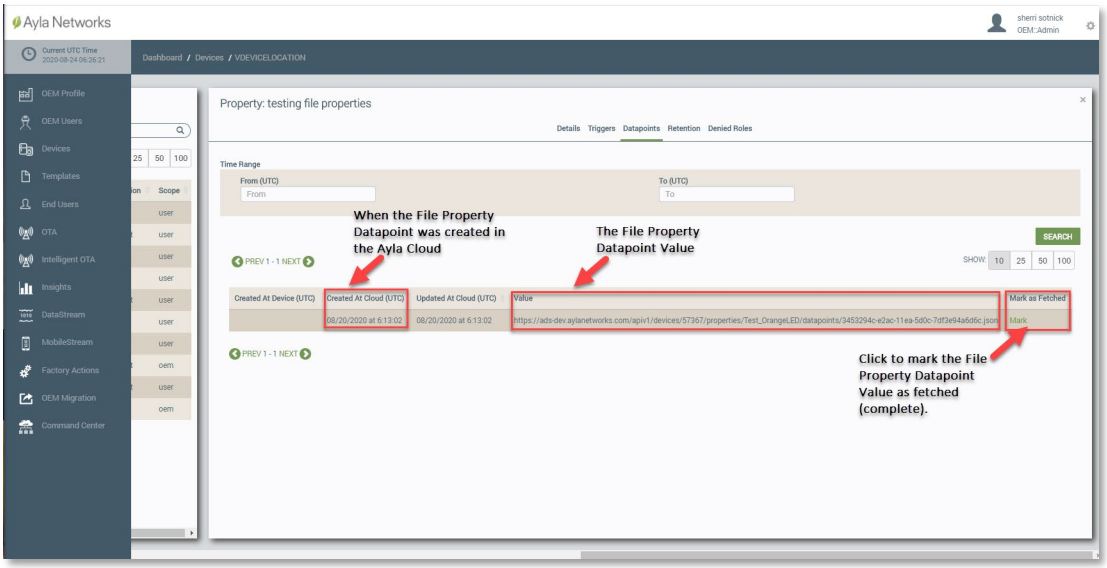
15. Click **OK** in the confirmation message box.

You can also manage and mark your file properties as fetched in the Ayla Customer Dashboard as follows:

1. After creating your new file property (Step 11 above), navigate to the Ayla Customer Dashboard, and log in.
2. Click **Devices** in the navigation panel
3. Find the device to which you are either uploading or downloading the file, and then click the device.
4. Click **PROPERTIES**, and then find and click your new file property in the list. Refer to the following example:



5. Click the **Datapoints** tab. In the following example, notice that there is a date/time stamp for when the file property datapoint was created in the Ayla Cloud, and that the datapoint value is a location URL in a JSON data-interchange format.



Also notice in the example above that this is where you mark the file property datapoint as fetched (completed) once the file is downloaded. This is a mandatory step as explained in the File Property Dataflow section.

4 Additional Resources for File Properties

- Refer to the *Ayla Module and MCU Interface Guide* and the *Ayla Demo API Guide* for more details on file property operations in the Ayla Embedded Agent.
- Refer to the *Ayla Embedded Agent for Embedded Systems Guide* for details on the file property datapoint structure, such as descriptions of the attributes and functions, including how they are used in handling the file properties.

680 N. McCarthy Blvd., Suite 100
Milpitas, CA 95054
Phone: +1 408 830 9844
Fax: +1 408 716 2621