

ADLINK Technical Document

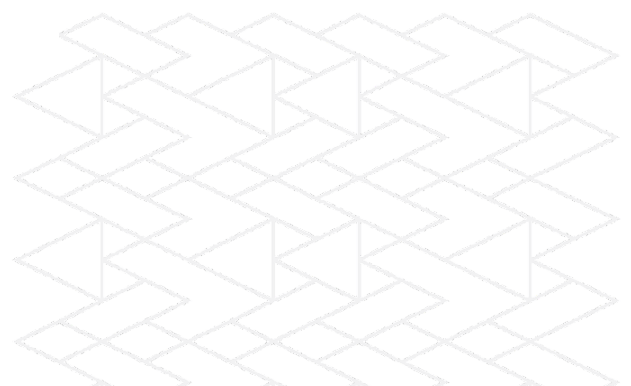
Abstract	How to Build a Customized Algorithm on Linux		
OS	Linux		
Keyword	Linux, Customized Algorithm		
Related Products	MCM-204, MCM-216, MCM-218		
Date	2021-10-29	No.	202110001

- Issue Details:

The MCM-204, MCM-216, and MCM-218 DAQ allow the deployment of custom algorithms for different applications. The MCM-204, MCM-216, and MCM-218 are ARM-based systems that require cross-compilers to compile customized algorithms. This document shows how to build and deploy a customized algorithm on the MCM-204, MCM-216, and MCM-218.

- Prerequisites:

Install Ubuntu 18.04 OS on any PC.



- Solution:

Step 1: Download ARM GCC Tool

Download the GNU Arm Embedded toolchain compiler GCC.

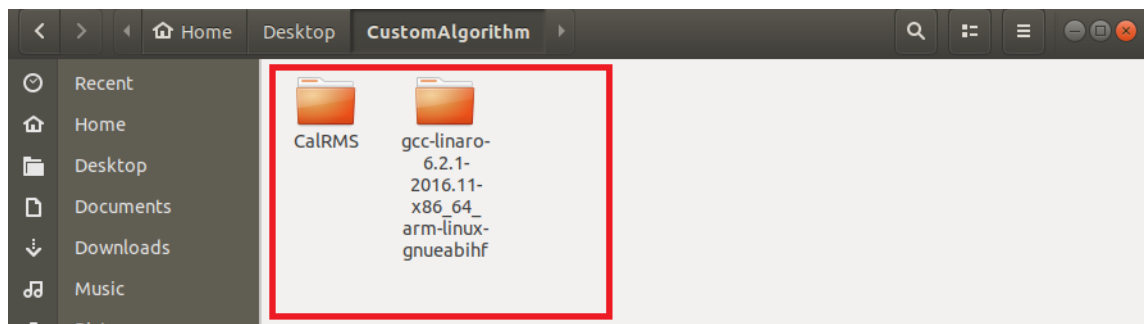
1. Go to <https://releases.linaro.org/components/toolchain/binaries/6.2-2016.11/arm-linux-gnueabi/>
2. Download **gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz**

Name	Last modified	Size	License
Parent Directory			
gcc-linaro-6.2-2016.11.tar.xz	27-Feb-2018 00:27	75.1M	open
gcc-linaro-6.2-2016.11.tar.xz.asc	13-Jan-2018 07:10	64	open
gcc-linaro-6.2.1-2016.11-i686-mingw32_arm-linux-gnueabi.tar.xz	27-Feb-2018 00:27	250.3M	open
gcc-linaro-6.2.1-2016.11-i686-mingw32_arm-linux-gnueabi.tar.xz.asc	13-Jan-2018 07:10	99	open
gcc-linaro-6.2.1-2016.11-i686_arm-linux-gnueabi.tar.xz	27-Feb-2018 00:27	87.6M	open
gcc-linaro-6.2.1-2016.11-i686_arm-linux-gnueabi.tar.xz.asc	13-Jan-2018 07:11	91	open
gcc-linaro-6.2.1-2016.11-linux-manifest.txt	13-Jan-2018 07:11	7.1K	open
gcc-linaro-6.2.1-2016.11-win32-manifest.txt	13-Jan-2018 07:11	7.6K	open
gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz	27-Feb-2018 00:28	87.3M	open
gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz.asc	13-Jan-2018 07:11	93	open
runtime-gcc-linaro-6.2.1-2016.11-arm-linux-gnueabi.tar.xz	13-Jan-2018 07:11	6.2M	open

3. Extract the downloaded file

Step 2: Download Sample Files

1. Download the customized algorithm sample file from the ADLINK website.
 - MCM-204: <https://ftp.adlinktech.com/daq/MCM-204.zip>
 - MCM-216: <https://ftp.adlinktech.com/daq/MCM-216.zip>
2. Extract the downloaded file.
3. Copy files to the same folder as the GCC tool.



Step 3: Build library (.so file)

1. Install the **make** tool if needed (most Linux distros include the **make** tool by default).

```
$ sudo apt install make
```

```
mcm200@ubuntu:~/Desktop/CustomAlgorithm$ sudo apt install make
[sudo] password for mcm200:
Reading package lists... Done
Building dependency tree
Unpacking make (4.1-9.1ubuntu1) ...
Setting up make (4.1-9.1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
mcm200@ubuntu:~/Desktop/CustomAlgorithm$
```

2. Change the directory to “CalRMS” and use the **make** command to build the .so file from the sample.

```
$ cd CalRMS
$ make
```

```
mcm200@ubuntu:~/Desktop/CustomAlgorithm$ cd CalRMS/
mcm200@ubuntu:~/Desktop/CustomAlgorithm/CalRMS$ make
/home/mcm200/Desktop/CustomAlgorithm/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc -c -fPIC -Wextra -O2 -g customAlgo.c
customAlgo.c: In function 'CustomAlgo':
customAlgo.c:40:22: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
    for(int i = 0; i < devInfo.dataCount; i++)
                       ^
/home/mcm200/Desktop/CustomAlgorithm/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc -shared -o customAlgo.so customAlgo.o
mcm200@ubuntu:~/Desktop/CustomAlgorithm/CalRMS$
```

3. Enter the **ls** command to list all files and check if **customAlgo.so** exists in the current directory.

```
$ ls
```

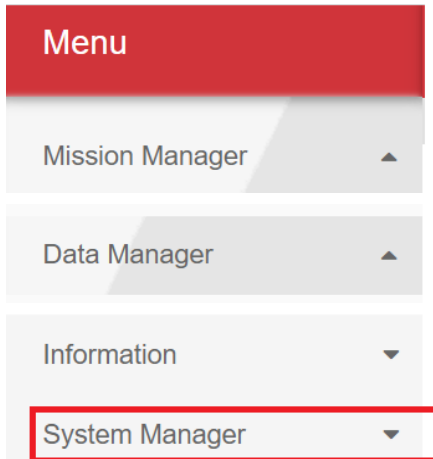
```
mcm200@ubuntu:~/Desktop/CustomAlgorithm/CalRMS$ ls
customAlgo.c customAlgo.h customAlgo.o customAlgo.so Makefile
mcm200@ubuntu:~/Desktop/CustomAlgorithm/CalRMS$
```

Step 4: Deploy the library (customAlgo.so file) to MCM-204

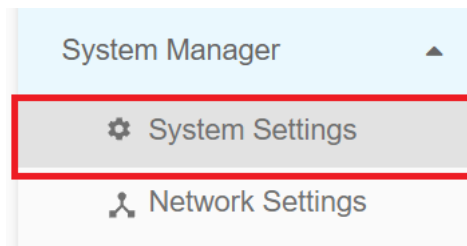
The **customAlgo.so** file created in the previous step must be deployed to the DAQ device.



1. Log in to the MCM-204 Web Console.
2. In the menu, click **System Manager**.



3. Click **System Settings**

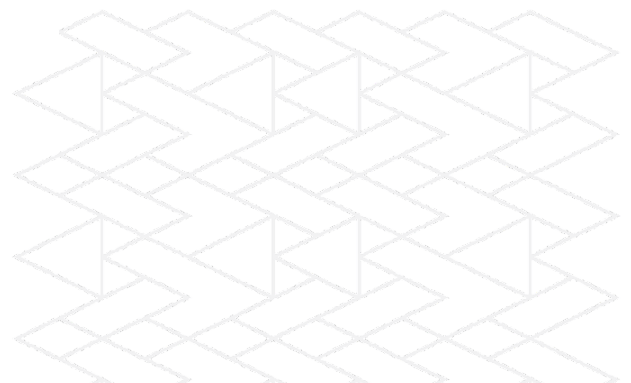


4. Scroll down to the **Customization library Upload** section, then click **Choose File**.

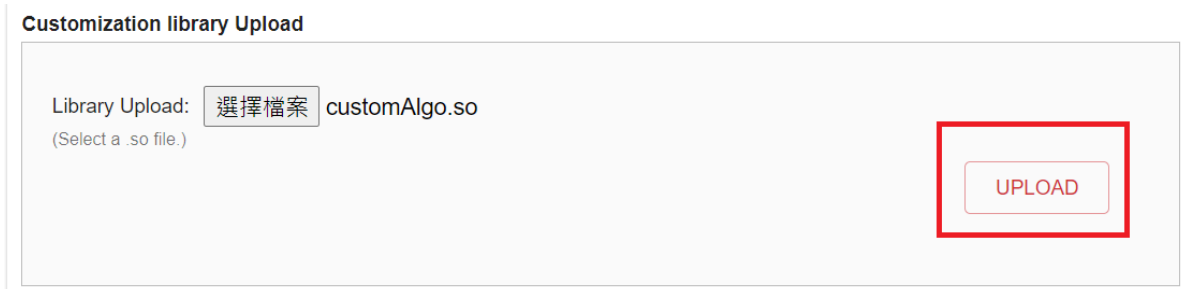
Customization library Upload



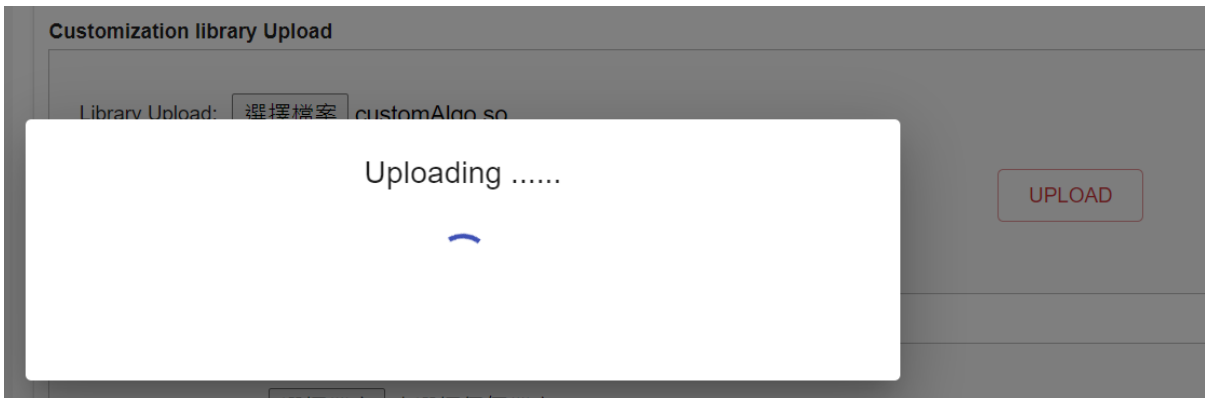
The screenshot shows the 'Customization library Upload' section. It contains a 'Library Upload:' label with a sub-label '(Select a .so file.)'. To the right of the label is a text input field containing the Chinese text '選擇檔案' (Choose file) and '未選擇任何檔案' (No file selected). The input field is highlighted with a red rectangular box. To the right of the input field is an 'UPLOAD' button.



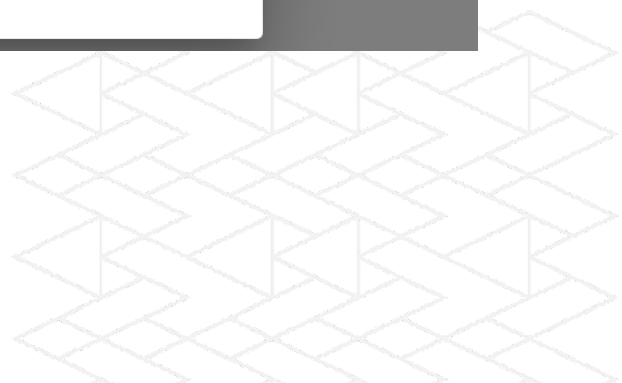
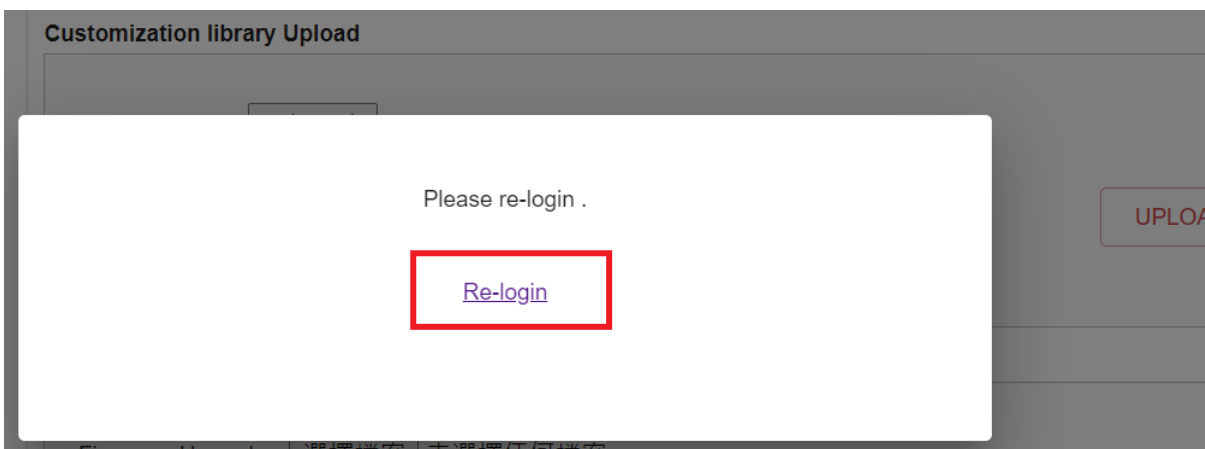
5. Select **customAlgo.so**, then click **UPLOAD** to upload the file to the DAQ.



6. Wait about 30 seconds for the upload process to complete.



7. Click the link to log in again when prompted.

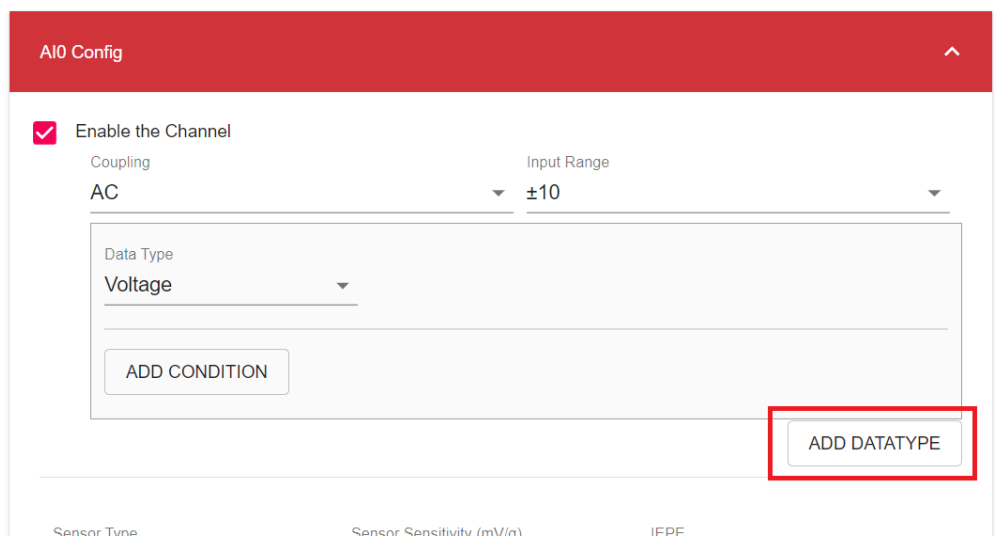


Step 5: Configure Analog Input

As an example, we'll configure settings for AIO.

1. Click **Device Settings**.
2. Scroll down to the **Channel Config** section.
3. Click **AIO Config** to open the AIO configuration.
4. Check the **Enable the Channel** checkbox.
5. Click **ADD DATATYPE**.

Channel Config



AIO Config

Enable the Channel

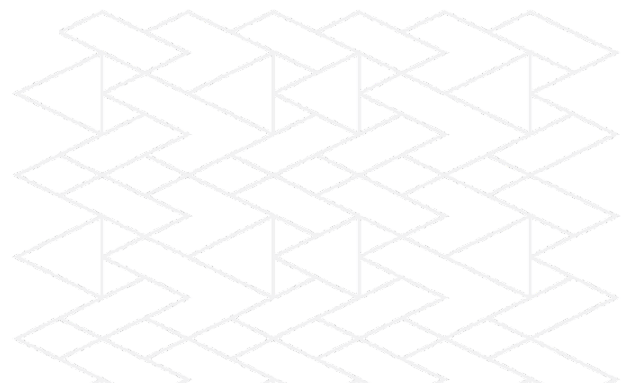
Coupling: AC Input Range: ±10

Data Type: Voltage

ADD CONDITION

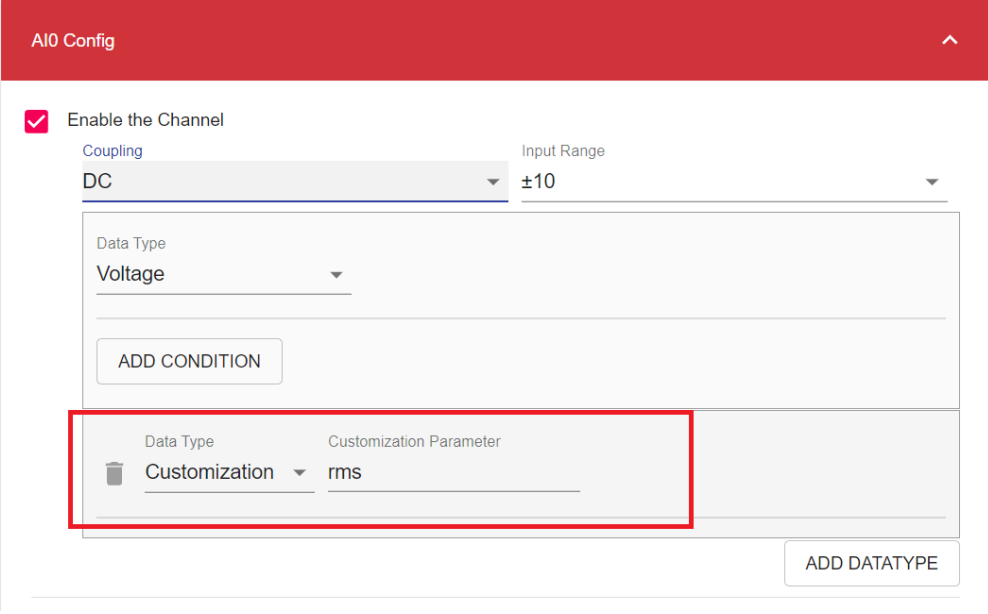
ADD DATATYPE

Sensor Type Sensor Sensitivity (mV/in) IFPE



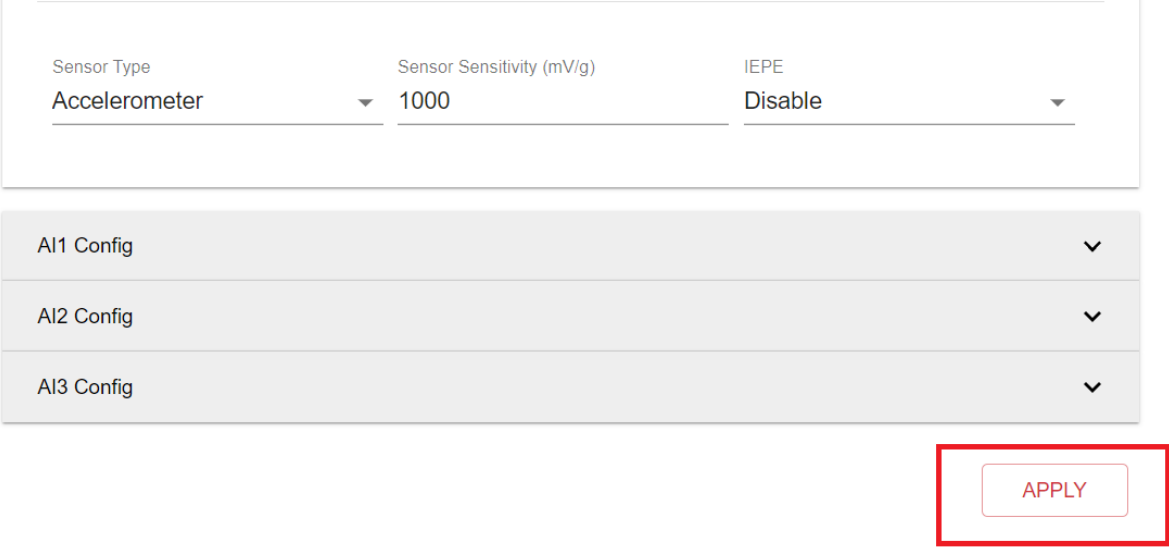
6. Set the new condition parameters:

- For **Date Type**, select **Customization**
- For **Customization Parameter**, select **rms**

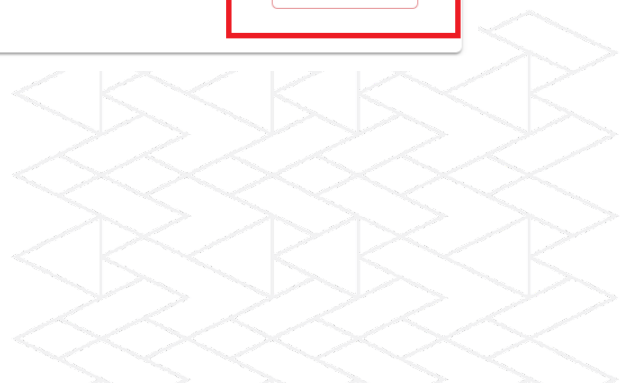


The screenshot shows the 'AI0 Config' window. At the top, there is a red header with 'AI0 Config' and an upward arrow. Below the header, there is a section titled 'Enable the Channel' with a checked checkbox. Underneath, there are two dropdown menus: 'Coupling' set to 'DC' and 'Input Range' set to '±10'. Below these is a larger dropdown menu for 'Data Type' set to 'Voltage'. An 'ADD CONDITION' button is located below the 'Data Type' dropdown. A red box highlights a row in a table below, with columns 'Data Type' and 'Customization Parameter'. The row contains 'Customization' and 'rms'. An 'ADD DATATYPE' button is located at the bottom right of the table area.

7. Scroll down to the bottom of the page and click **APPLY**.




The screenshot shows the bottom of the configuration page. It features three dropdown menus: 'Sensor Type' set to 'Accelerometer', 'Sensor Sensitivity (mV/g)' set to '1000', and 'IEPE' set to 'Disable'. Below these are three rows for 'AI1 Config', 'AI2 Config', and 'AI3 Config', each with a downward arrow. At the bottom right, there is a red box containing an 'APPLY' button.



Step 6: Check Data

Click **Data Capture** and the data will display under the **Customization** key in the returned JSON data.

Data In JSON 

```
{ 2 items
  "Data": [ 2 items
    0: { 1 item
      "AI0": { 2 items
        "Voltage": 10240 items
          [ 0 - 1000 ]
          [ 1000 - 2000 ]
          [ 2000 - 3000 ]
          [ 3000 - 4000 ]
          [ 4000 - 5000 ]
          [ 5000 - 6000 ]
          [ 6000 - 7000 ]
          [ 7000 - 8000 ]
          [ 8000 - 9000 ]
          [ 9000 - 10000 ]
          [ 10000 - 10240 ]
          "Customization": [ 1 item
            0: 0.00031754674
          ]
        }
      }
    }
    1: { 1 item
      "Date": "2019-11-05 14:27:34.016"
    }
  ]
  "Date": "2019-11-05 14:27:34.016"
}
```

