WebAccess MQTT
Design Specification for Device

Version：1.2.1

Author：Oz.Chen, Steven.Li
Advantech Co., Ltd
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Editor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>2016/2/1</td>
<td>Oz.Chen</td>
<td>First</td>
</tr>
<tr>
<td>1.0.1</td>
<td>2016/2/17</td>
<td>Oz.Chen</td>
<td>Add default value for Tag Common Part</td>
</tr>
<tr>
<td>1.0.2</td>
<td>2016/3/18</td>
<td>Oz.Chen</td>
<td>Add file type for file transfer</td>
</tr>
<tr>
<td>1.0.3</td>
<td>2016/3/23</td>
<td>Oz.Chen</td>
<td>Add action format</td>
</tr>
<tr>
<td>1.0.4</td>
<td>2016/3/25</td>
<td>Oz.Chen</td>
<td>Add heartbeat interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Add primary device ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modify heartbeat</td>
</tr>
<tr>
<td>1.1.0</td>
<td>2016/4/13</td>
<td>Oz.Chen</td>
<td>Add data recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Add time synchronization</td>
</tr>
<tr>
<td>1.1.1</td>
<td>2016/4/15</td>
<td>Oz.Chen</td>
<td>Add get tag list with datalog or RTDB enabled</td>
</tr>
<tr>
<td>1.1.2</td>
<td>2016/5/6</td>
<td>Oz.Chen</td>
<td>Add device and tags delete all</td>
</tr>
<tr>
<td>1.1.3</td>
<td>2016/5/11</td>
<td>Oz.Chen</td>
<td>Modify data recovery format</td>
</tr>
<tr>
<td>1.1.4</td>
<td>2016/5/13</td>
<td>Oz.Chen</td>
<td>Modify time stamp format</td>
</tr>
<tr>
<td>1.1.5</td>
<td>2016/5/24</td>
<td>Oz.Chen</td>
<td>Remove data recovery tag list download request</td>
</tr>
<tr>
<td>1.1.6</td>
<td>2016/6/7</td>
<td>Oz.Chen</td>
<td>Remove PRI, add BID for configuration</td>
</tr>
<tr>
<td>1.1.7</td>
<td>2016/7/4</td>
<td>Oz.Chen</td>
<td>Modify data communication mechanism at start time</td>
</tr>
<tr>
<td>1.2.0</td>
<td>2017/9/27</td>
<td>Oz.Chen</td>
<td>Change Document Title</td>
</tr>
<tr>
<td>1.2.1</td>
<td>2017/11/17</td>
<td>Steven Li</td>
<td>Modify the wrong format of data recovery example</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Connect to MQTT Broker of WebAccess Cloud ........................................... 5
   1.1 Connection Parameter ........................................................................... 5
      1.1.1 Host Name & Port ........................................................................ 5
      1.1.2 Client ID ..................................................................................... 5
      1.1.3 QoS ........................................................................................... 6
      1.1.4 Username / Password .................................................................. 6
      1.1.5 SSL/TLS ..................................................................................... 7
      1.1.6 Conclusion .................................................................................. 7
   1.2 Topic Define ......................................................................................... 7
      1.2.1 Data ........................................................................................... 7
      1.2.2 File ............................................................................................ 7
      1.2.3 Command .................................................................................. 7
      1.2.4 Connection ................................................................................ 8
   1.3 Root Json Format .................................................................................. 8
2. MQTT Communication Format ..................................................................... 9
   2.1 Data Format ........................................................................................ 9
      2.1.1 Tag Value ................................................................................... 9
      2.1.2 The Rules of Device Data Communication ........................................ 12
      2.1.3 Data Recovery ............................................................................ 13
   2.2 File Format ........................................................................................ 15
      2.2.1 File Upload ................................................................................ 15
      2.2.2 File Download ............................................................................ 17
   2.3 Configuration Format .......................................................................... 19
      2.3.1 Introduction ................................................................................. 19
      2.3.2 Device ........................................................................................ 19
      2.3.3 Tag Common Part ....................................................................... 21
      2.3.4 Analog Tag Advance Part ............................................................. 23
      2.3.5 Digit Tag Advance Part ................................................................ 25
      2.3.6 Configuration Download ............................................................. 27
      2.3.7 Add Tag Example ....................................................................... 28
      2.3.8 Update Tag Example ................................................................... 32
      2.3.9 Delete Tag Example .................................................................... 33
      2.3.10 Devices Plug and Play Mechanism .............................................. 34
   2.4 Command Format ............................................................................... 35
      2.4.1 Write Value ................................................................................ 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.2 Write Config</td>
<td>37</td>
</tr>
<tr>
<td>2.4.3 Data On</td>
<td>39</td>
</tr>
<tr>
<td>2.4.4 Data Off</td>
<td>40</td>
</tr>
<tr>
<td>2.4.5 Backup Configuration</td>
<td>41</td>
</tr>
<tr>
<td>2.4.6 Restore Configuration</td>
<td>42</td>
</tr>
<tr>
<td>2.4.7 Firmware Upgrade</td>
<td>43</td>
</tr>
<tr>
<td>2.4.8 File Download</td>
<td>44</td>
</tr>
<tr>
<td>2.4.9 Time Synchronization</td>
<td>46</td>
</tr>
<tr>
<td>2.4.10 Command List</td>
<td>47</td>
</tr>
<tr>
<td>2.5 Connection Format</td>
<td>48</td>
</tr>
<tr>
<td>2.5.1 Connect</td>
<td>48</td>
</tr>
<tr>
<td>2.5.2 Disconnect</td>
<td>49</td>
</tr>
<tr>
<td>2.5.3 Unexpected Disconnect</td>
<td>50</td>
</tr>
<tr>
<td>2.5.4 Heartbeat</td>
<td>51</td>
</tr>
<tr>
<td>2.5.5 Time Synchronization</td>
<td>52</td>
</tr>
<tr>
<td>2.6 Action Format</td>
<td>53</td>
</tr>
<tr>
<td>2.6.1 Connect</td>
<td>53</td>
</tr>
<tr>
<td>2.6.2 Unexpected Disconnect</td>
<td>54</td>
</tr>
<tr>
<td>2.6.3 Config</td>
<td>55</td>
</tr>
<tr>
<td>2.6.4 File</td>
<td>56</td>
</tr>
</tbody>
</table>
1. Connect to MQTT Broker of WebAccess Cloud

Devices which want to connect to WebAccess Cloud must go with MQTT.

1.1 Connection Parameter

1.1.1 Host Name & Port

**Host Name:** Domain name or IP address with "/WaMQTT/" of MQTT broker
Ex: wacloud.appcloud.net/WaMQTT/

**Port:** Port of MQTT broker. Default port of WS(WebSocket), WSS(Secure WebSocket) and TCP are 80, 443, 1883. It is strongly recommended devices go with MQTT over WSS for security.

1.1.2 Client ID

Client ID must be unique for each device and the format is:

```
d:group_id:type_id:dev_id
```

1.1.2.1 d

It means devices.

1.1.2.2 group_id

`group_id` is composed by WebAccess Cloud project name and SCADA name. It is seperated by a under line, for example: MyProject_MySCADA.
The maximum length of `group_id` is 65, and each of project name and SCADA name is 32.

1.1.2.3 type_id

It means device type.
### 1.1.2.4 dev_id

It means device ID. The maximum length is 32. 

**dev_id must be unique under same group_id.**

### 1.1.3 QoS

The quality of service (QoS) level is an agreement between sender and receiver of a message regarding the guarantees of delivering a message.

#### 1.1.3.1 QoS0

At most once.

#### 1.1.3.2 QoS1

At least once. *(QoS1 is default suggestion)*

#### 1.1.3.3 QoS2

Exactly once. *(not support yet)*

### 1.1.4 Username / Password

User name and password for MQTT broker

<table>
<thead>
<tr>
<th>type_id</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default or Unknow</td>
</tr>
<tr>
<td>1</td>
<td>WebAccess/SCADA</td>
</tr>
<tr>
<td>2</td>
<td>WebAccess/HMI</td>
</tr>
<tr>
<td>3</td>
<td>WISE4000</td>
</tr>
<tr>
<td>4</td>
<td>ADAM3600</td>
</tr>
<tr>
<td>5</td>
<td>B+B Wzzard</td>
</tr>
</tbody>
</table>
1.5 SSL/TLS

WebAccess Cloud MQTT broker is support SSL/TLS for WSS only.

1.6 Conclusion

With those connection parameters for WebAccess Cloud MQTT broker, there must be some UI for user to input relative information on the devices.

1.2 Topic Define

There are several MQTT topic as the following:

1.2.1 Data

iot-2/evt/wdata/fmt/group_id
This topic is for devices sending their real-time data.

1.2.2 File

iot-2/evt/wfile/fmt/group_id
This topic is for devices sending their file.

1.2.2 Configuration

iot-2/evt/wcfg/fmt/group_id
This topic is for devices sending their configuration.

1.2.3 Command

iot-2/evt/wacmd/fmt/group_id
This two topics are for devices receiving command from cloud.

1.2.4 Connection

This topic is for devices sending their connection status.

1.3 Root Json Format

There are only two property in root json format.

Format:

```
{
    "d": {
        "ts": "2015-04-04T23:26:10+08:00"
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Json Format</td>
<td>User define data</td>
</tr>
</tbody>
</table>
| ts   | UTC Time stamp, support format as the following:  
1. 2016-05-13T02:52:51Z (seconds)  
2. 2016-05-13T02:52:51.742Z (milliseconds)  
3. 2016-05-13T10:52:51+08:00 (seconds with time zone)  
4. 2016-05-13T10:52:51.742+08:00 (milliseconds with time zone) |
2. MQTT Communication Format

In this chapter, it will define json format of data, file, configuration, command and connection topics.

There are several representation styles as the following:
- Red means primary key
- Green means necessary
- Blue means optional but nice to have
- Black means optional
- ... means user define config space

It is case sensitive for all json property.

2.1 Data Format

Devices publish real-time data to this topic:

```
iot-2/evt/wdata/fmt/group_id
```

2.1.1 Tag Value

It is important for saving bandwidth that devices should publish real-time data which the tag values are changed. It is recommended to have deadband setting for each tag when difference between current and last published value is over some percentage.

Format:

```
{
   "unique string": {
      "Val": {
         "unique string1": number,
         "unique string2": "string",
         "unique string3": {
            "0": number,
            "1": number,
            "2": number,
            ...
         }
      }
   },
   ...
}
```
Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unique string</strong></td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td><strong>unique string1~3</strong></td>
<td>21</td>
<td>Tag ID, must be unique under same WebAccess project and can not be &quot;(&quot; , &quot;,&quot;, &quot;,&quot;, &quot;:&quot;, &quot;.&quot;, &quot;,$&quot;, &quot;,=&quot;, &quot;,#&quot; and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address) Tag Value can be number, string or array (index from 0). One single &quot;:&quot; is for bad value. The value should be bad when devices or I/O can not be reached.</td>
</tr>
</tbody>
</table>

*The maximum length of WebAccess Tag Name is limited to 21 bytes.

Example for normal value:

```json
{
  "d": {
    "WISE4010-7F28A7": {
      "Val": {
        "W4010-28A7_Fz1Volt": 1213.48,
        "W4010-28A7_Fz1Temp": 63.81,
        "W4010-28A7_Fz1Desc": "Freezer 1",
      }
    }
  },
  "ts": "2015-04-17T10:51+08:00"
}
```
Example for array value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Val": {
                "W4010-28A7_Fz1Volt": {
                    "0": 963.56,
                    "3": 1147.38,
                    "4": 1038.84
                }
            }
        },
        "ts": "2015-04-17T10:19:51+08:00"
    }
}
```

Example for bad value:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "Val": {
                "W4010-28A7_Fz1Volt": "$",
                "W4010-28A7_Fz1Temp": "$",
                "W4010-28A7_Fz1Desc": "$"
            }
        },
        "ts": "2015-04-17T10:19:51+08:00"
    }
}
```
2.1.2 The Rules of Device Data Communication

It supports primary and backup devices mechanism on WebAccess Cloud. The backup devices is switched to data communication with cloud when the primary devices are lost. And it will switch back to primary when the primary devices is come back.

For saving network bandwidth(2.1.1):
- All devices publish real-time data to cloud when value of tags is changed.

For primary and backup devices(2.3.2):
- All devices can not publish real-time data at the start.

For command of data on(2.4.4) and data off(2.4.5):
- The devices should publish real-time data when receiving command of data on, and publish all values only at first time.
- The devices should stop publishing real-time data when receiving command of data off.
2.1.3 Data Recovery

Cloud sides can collect data from devices and record those to database for query or analytics usage. However, it gets data lost when devices is offline unexpectedly. Therefore, It will be necessary to recover the lost data as the following when devices found that they didn’t publish some period of data successfully.

Format:

```json
{
    "unique string": {
        "DRec": {
            "From": number,
            "Tags": {
                "unique string1": {
                    "offset1": number,
                    "offset2": number,
                    ...}
            },
            "unique string2": {
                "offset1": "string",
                "offset2": "string",
                ...}
        }
    }
}
```
### Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unique string</strong></td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td><strong>DRec</strong></td>
<td></td>
<td>Data Recovery</td>
</tr>
<tr>
<td><strong>From</strong></td>
<td></td>
<td>Seconds since 00:00 hours, Jan 1, 1970 UTC</td>
</tr>
<tr>
<td><strong>unique string1~N</strong></td>
<td>21</td>
<td>Tag ID, must be unique under same WebAccess project and can not be &quot;(&quot; , &quot;,&quot;, &quot;,&quot;, &quot;,&quot;: &quot;,&quot;: &quot;,&quot;: &quot;,&quot;: &quot;,&quot;: &quot; and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address) Tag Value can be number, string or array (index from 0). One single &quot;*&quot; is for bad value. The value should be bad when devices or I/O can not be reached.</td>
</tr>
<tr>
<td><strong>offset 1~N</strong></td>
<td></td>
<td>Time Offset with <strong>From</strong> in seconds. Value can be number, string or array (index from 0)</td>
</tr>
</tbody>
</table>

#### Example:

```json
{
    "d": {
        "WISE4010-7F28A7": {
            "DRec": {
                "From": 1460129890,
                "Tags": {
                    "W4010-28A7_Fz1Volt": {
                        "3": 147.32,
                        "18": 149.18,
                        "87": 148.93
                    },
                    "W4010-28A7_Fz2Volt": {
                        "0": 73.83,
                        "118": 84.49,
                        "124": 68.17
                    }
                }
            }
        }
    }
}
```

"ts": "2015-04-17T10:19:51+08:00"
2.2 File Format

Devices publish file to this topic:
\texttt{iot-2/evt/wafie/fmt/group\_id}

2.2.1 File Upload

Upload file from devices to cloud

Format:

```
{  
  "unique string": {  
    "FPut": {  
      "Name": "string",  
      "Ind": number,  
      "Type": number,  
      "Size": number,  
      "Bin": binary  
    }  
  },  
  ...  
}
```
Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unique string</strong></td>
<td>31</td>
<td>Device ID (unique under same $\text{group_id}$)</td>
</tr>
<tr>
<td><strong>FPut</strong></td>
<td></td>
<td>File Upload to Cloud</td>
</tr>
<tr>
<td>Name</td>
<td>256</td>
<td>{File Path}/{File Name}</td>
</tr>
<tr>
<td><strong>Ind</strong></td>
<td>0 or 1</td>
<td>Is Individual for Root Path:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: …\WebAccess\Node\mqtt\</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: …\WebAccess\Node\config{group_id}</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>0 ~ 2</td>
<td>0: ASCII</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: UTF-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Binary</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>&lt;=256MB</td>
<td>File Size In Bytes (Up to 4GB)</td>
</tr>
<tr>
<td><strong>Bin</strong></td>
<td></td>
<td>File Content</td>
</tr>
</tbody>
</table>

Example for file upload:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "FPut": {
                "Name": "W4010-28A7/image.bin",
                "Ind": 1,
                "Type": 2,
                "Size": 3576,
                "Bin": "content…"
            }
        }
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```
2.2.2 File Download

Download file from cloud to device

Format:

```json
{
    "unique string": {
        "FGet": {
            "Name": "string",
            "Ind": number
        }
    },
    ...
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same <code>group_id</code>)</td>
</tr>
<tr>
<td>FGet</td>
<td></td>
<td>File Download from Cloud</td>
</tr>
<tr>
<td>Name</td>
<td>256</td>
<td>{File Path}/{File Name}</td>
</tr>
<tr>
<td>Ind</td>
<td>0 or 1</td>
<td>Is Individual for Root Path:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: …/WebAccess/Node/mqtt\</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: …/WebAccess/Node/config/{group_id} \</td>
</tr>
</tbody>
</table>
Example for file download:

```json
{
   "d": {
      "WISE4010-7F28A7": {
         "FGet": {
            "Name": "W4010-28A7/image.bin",
            "Ind": 1
         }
      },
      "ts": "2015-04-17T10:19:51+08:00"
   }
}
```

(file download response from command "Fdl" at 2.4.8)
2.3 Configuration Format

Devices publish configuration to this topic:

```
iot-2/evt/wacfg/fmt/group_id
```

2.3.1 Introduction

On WebAccess Cloud, it has hierarchy configuration as the following:

```
+Project
    +SCADA
        +Port
            +Device
                +Tag
```

It can be configured automatically on WebAccess Cloud if devices publish sufficient configuration information. Devices should keep a last uploaded configuration profile, and sending differential part to cloud every time by "UTg" or "DTg". In this way, devices can add, modify or delete tag configuration on WebAccess Cloud.

2.3.2 Device

Format:

```
{
    "unique string": {
        "TID": number,
        "Dsc": "string",
        "Hbt": number,
        "BID": "string",
        "UTg": {tag format},
        "DTg": {tag format},
        "Del": 1,
        ...
    }
}
```
### Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unique string</strong></td>
<td>31</td>
<td>Device ID (unique under same group.id)</td>
</tr>
<tr>
<td>TID</td>
<td>1~5</td>
<td>Device Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: WebAccess/SCADA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: WebAccess/HMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: WISE4000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4: ADAM3600</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. B+B Wzzard</td>
</tr>
<tr>
<td><strong>Dsc</strong></td>
<td>64</td>
<td>Description</td>
</tr>
<tr>
<td>Hbt</td>
<td>1~65535</td>
<td>Interval of Heartbeat (2.5.4), Suggest 5 or 10 Seconds Should be Good</td>
</tr>
<tr>
<td>BID</td>
<td>31</td>
<td>Backup Device ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left backup device ID if the device is backup</td>
</tr>
<tr>
<td><strong>UTg</strong></td>
<td></td>
<td>Add or Update Tag</td>
</tr>
<tr>
<td><strong>DTg</strong></td>
<td></td>
<td>Delete Tag</td>
</tr>
<tr>
<td><strong>Del</strong></td>
<td>1</td>
<td>Delete Device and All Tags on Cloud</td>
</tr>
</tbody>
</table>

PS: Command order is "**Del**" → "**DTg**" → "**UTg**" if they show at same time
2.3.3 Tag Common Part

Format:

"unique string": {
    "TID": number,
    "Dsc": "string",
    "Ary": number,
    "RO": number,
    ...  
}

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>21</td>
<td>Tag ID, must be unique under same WebAccess project and can not be &quot;(&quot; , &quot;,&quot; , &quot;,&quot;, &quot;,&quot;, &quot;:&quot;, &quot;,&quot;, &quot;:&quot;, &quot;,&quot;, &quot;:&quot;, &quot;,&quot;, &quot;:&quot;, &quot;,&quot;, &quot;:&quot;, &quot;,&quot;, &quot;:&quot;, &quot;%&quot; , &quot;=&quot; , &quot;#&quot; and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address)</td>
<td></td>
</tr>
<tr>
<td>TID</td>
<td>1~3</td>
<td>Tag Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Analog, 2: Digit, 3: Text</td>
<td></td>
</tr>
<tr>
<td>Dsc</td>
<td>64</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Ary</td>
<td>0~32767</td>
<td>Array Size</td>
<td>0</td>
</tr>
<tr>
<td>RO</td>
<td>0 or 1</td>
<td>Read Only (0: false, 1: true)</td>
<td>0</td>
</tr>
</tbody>
</table>

*The maximum length of WebAccess Tag Name is limited to 21 bytes.
2.3.3.1 Tag ID

It is recommended to use abbreviation because the length is limitation. WebAccess Cloud is tag based architecture so it is suggested that using the format as the following will be good:

{Device Unique Abbreviation}_{Tag Abbreviation}

For example, there is a WISE4012E with partial MAC address A924 and one of analog input channel can be show:

W4012-A924_AI0

Notice that devices should keep Tag ID to be unique under the same WebAccess project.
2.3.4 Analog Tag Advance Part

Format:

```json
{
    "Log": number,
    "SH": number,
    "SL": number,
    "EU": "string",
    "DSF": "string",
    "Alm": number,
    "HHP": number,
    "HHA": number,
    "HiP": number,
    "HiA": number,
    "LoP": number,
    "LoA": number,
    "LLP": number,
    "LLA": number,
    ...  
}
```
### Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>0 or 1</td>
<td>Log Data to RTDB (0:disabled, 1:enabled), RTDB means WebAccess Real-time Database</td>
<td>0</td>
</tr>
<tr>
<td>SH</td>
<td>double</td>
<td>Span High</td>
<td>1000</td>
</tr>
<tr>
<td>SL</td>
<td>double</td>
<td>Span Low</td>
<td>0</td>
</tr>
<tr>
<td>EU</td>
<td>10</td>
<td>Engineer Unit</td>
<td></td>
</tr>
<tr>
<td>DSF</td>
<td>xx.xx</td>
<td>Display Format (Integer.Fraction) (xx: 0~15)</td>
<td>4.2</td>
</tr>
<tr>
<td>Alm</td>
<td>0 or 1</td>
<td>Alarm Enabled (0:false, 1:true)</td>
<td>0</td>
</tr>
<tr>
<td>HHP</td>
<td>0~99</td>
<td>HH Priority (0:disabled)</td>
<td>0</td>
</tr>
<tr>
<td>HHA</td>
<td>double</td>
<td>HH Alarm Limit</td>
<td></td>
</tr>
<tr>
<td>HiP</td>
<td>0~99</td>
<td>High Priority (0:disabled)</td>
<td>0</td>
</tr>
<tr>
<td>HiA</td>
<td>double</td>
<td>High Alarm Limit</td>
<td></td>
</tr>
<tr>
<td>LoP</td>
<td>0~99</td>
<td>Low Priority (0:disabled)</td>
<td>0</td>
</tr>
<tr>
<td>LoA</td>
<td>double</td>
<td>Low Alarm Limit</td>
<td></td>
</tr>
<tr>
<td>LLP</td>
<td>0~99</td>
<td>LL Priority (0:disabled)</td>
<td>0</td>
</tr>
<tr>
<td>LLA</td>
<td>double</td>
<td>LL Alarm Limit</td>
<td></td>
</tr>
</tbody>
</table>

**Note1:** HHP >= HiP, LLP >= LoP  
**Note2:** Alm must be taken at same time when some relative alarm setting is updated
2.3.5 Digit Tag Advance Part

Format:

```json
{
    "Log": number,
    "S0": "string",
    "S1": "string",
    "S2": "string",
    "S3": "string",
    "S4": "string",
    "S5": "string",
    "S6": "string",
    "S7": "string",
    "Alm": number,
    "S0L": number,
    "S1L": number,
    "S2L": number,
    "S3L": number,
    "S4L": number,
    "S5L": number,
    "S6L": number,
    "S7L": number,
    ...
}
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>0 or 1</td>
<td>Log Data</td>
<td>0</td>
</tr>
<tr>
<td>S0</td>
<td>12</td>
<td>State 0</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>12</td>
<td>State 1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td>12</td>
<td>State 2</td>
<td>NotUsed</td>
</tr>
<tr>
<td>S3</td>
<td>12</td>
<td>State 3</td>
<td>NotUsed</td>
</tr>
<tr>
<td>S4</td>
<td>12</td>
<td>State 4</td>
<td>NotUsed</td>
</tr>
<tr>
<td>S5</td>
<td>12</td>
<td>State 5</td>
<td>NotUsed</td>
</tr>
<tr>
<td>S6</td>
<td>12</td>
<td>State 6</td>
<td>NotUsed</td>
</tr>
<tr>
<td>S7</td>
<td>12</td>
<td>State 7</td>
<td>NotUsed</td>
</tr>
<tr>
<td>Alm</td>
<td>0 or 1</td>
<td>Alarm Enabled (0:false, 1:true)</td>
<td>0</td>
</tr>
<tr>
<td>S0P</td>
<td>0~99</td>
<td>State 0 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S1P</td>
<td>0~99</td>
<td>State 1 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S2P</td>
<td>0~99</td>
<td>State 2 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S3P</td>
<td>0~99</td>
<td>State 3 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S4P</td>
<td>0~99</td>
<td>State 4 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S5P</td>
<td>0~99</td>
<td>State 5 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S6P</td>
<td>0~99</td>
<td>State 6 Alarm Priority</td>
<td>0</td>
</tr>
<tr>
<td>S7P</td>
<td>0~99</td>
<td>State 7 Alarm Priority</td>
<td>0</td>
</tr>
</tbody>
</table>
2.3.6 Configuration Download

Devices should keep a up-to-date configuration profile. If devices miss or mess up the profile, it can be restored by this request command. For example: device replacement, the new one need to fill up the device ID of the broken one, then using this request command to get back up-to-date configuration profile from cloud.

Format:

```json
{
    "unique string": {
        "Cdl": 1
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td>Cdl</td>
<td>1</td>
<td>Config Download</td>
</tr>
</tbody>
</table>

It will return configuration from 2.4.2 after cloud receive this request commnad.

Example:

```json
{
    "d": {
        "WISE4010-7F28A7": {
            "Cdl": 1
        }
    },
    "ts": "2015-04-17T10:51+08:00"
}
```
2.3.7 Add Tag Example

```json
{
  "d": {
    "WISE4010-7F28A7": {
      "TID": 3,
      "Dsc": "Factory A",
      "Hbt": 5,
      "PRI": "",
      "UTg": {
        "W4010-28A7_Fz1Volt": {
          "TID": 1,
          "Dsc": "Freezer1 Voltage",
          "RO": 1,
          "Log": 1,
          "SH": 2500,
          "SL": 0,
          "EU": "V",
          "DSF": "4.2",
          "Alm": 1,
          "HHP": 0,
          "HHA": 0,
          "HiP": 1,
          "HiA": 1800,
          "LoP": 1,
          "LoA": 500,
          "LLP": 0,
          "LLA": 0
        },
        "W4010-28A7_Fz2Volt": {
          "TID": 1,
          "Dsc": "Freezer2 Voltage",
          "RO": 1,
          "Log": 1,
          "SH": 2500,
          "SL": 0,
          "EU": "V",
          "DSF": "4.2",
        }
      }
    }
  }
}
```
"Alm": 1,
"HHP": 0,
"HHA": 0,
"HiP": 1,
"HiA": 1800,
"LoP": 1,
"LoA": 500,
"LLP": 0,
"LLA": 0
},
"W4010-28A7_Fz1Temp": {
  "TID": 1,
  "Dsc": "Freezer1 Temperature",
  "RO": 1,
  "Log": 1,
  "SH": 150,
  "SL": -50,
  "EU": "°C",
  "DSF": "3.2",
  "Alm": 1,
  "HHP": 0,
  "HHA": 0,
  "HiP": 1,
  "HiA": 90,
  "LoP": 1,
  "LoA": -10,
  "LLP": 0,
  "LLA": 0
},
"W4010-28A7_Fz2Temp": {
  "TID": 1,
  "Dsc": "Freezer2 Temperature",
  "RO": 1,
  "Log": 1,
  "SH": 150,
  "SL": -50,
  "EU": "°C",
  "DSF": "3.2"}
"DSF": "3.2",
"Alm": 1,
"HHP": 0,
"HHA": 0,
"HiP": 1,
"HiA": 90,
"LoP": 1,
"LoA": -10,
"LLP": 0,
"LLA": 0,

},
"W4010-28A7_Fz1Switch": {
  "TID": 2,
  "Dsc": "Freezer1 Switch",
  "RO": 0,
  "Log": 1,
  "S0": "On",
  "S1": "Off",
  "S2": "",
  "S3": "",
  "S4": "",
  "S5": "",
  "S6": "",
  "S7": "",
}
},
"W4010-28A7_Fz1Mode": {
  "TID": 2,
  "Dsc": "Freezer1 Mode Switch",
  "RO": 0,
  "Log": 1,
  "S0": "Save",
  "S1": "Safe",
  "S2": "Lv1",
  "S3": "Lv2",
  "S4": "Lv3",
  "S5": "Manual",
  "S6": "",
  "S7": "",
}
"W4010-28A7_Fz2Switch": {
    "TID": 2,
    "Dsc": "Freezer2 Switch",
    "RO": 0,
    "Log": 1,
    "S0": "On",
    "S1": "Off",
    "S2": "",
    "S3": "",
    "S4": "",
    "S5": "",
    "S6": "",
    "S7": ""
},
"W4010-28A7_Fz2Mode": {
    "TID": 2,
    "Dsc": "Freezer2 Mode Switch",
    "RO": 0,
    "Log": 1,
    "S0": "Save",
    "S1": "Safe",
    "S2": "Lv1",
    "S3": "Lv2",
    "S4": "Lv3",
    "S5": "Manual",
    "S6": "",
    "S7": ""
},
"ts": "2015-04-17T10:19:51+08:00"
2.3.8 Update Tag Example

```json
{
   "d": {
      "WISE4010-7F28A7": {
         "TID": 3,
         "Dsc": "Factory A",
         "Hbt": 5,
         "PRI": "",
         "UTg": {
            "W4010-28A7_Fz1Volt": {
               "SH": 5000,
               "SL": 500,
            },
            "W4010-28A7_Fz2Volt": {
               "SH": 7500,
               "SL": 2000,
            }
         }
      }
   },
   "ts": "2015-04-17T10:19:51+08:00"
}
```
2.3.9 Delete Tag Example

```json
{
    "d": {
        "WISE4010-7F28A7": {
            "TID": 3,
            "Dsc": "Factory A",
            "Hbt": 5,
            "PRI": ",",
            "DTg": {
                "W4010-28A7_Fz2Temp": 1,
                "W4010-28A7_Fz2Switch": 1
            }
        }
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```
2.3.10 Devices Plug and Play Mechanism

It still need to setup project, SCADA, ports, devices and tags because there is also a WebAccess software on cloud side. To reduce this configuration effort, auto-configuration from each device is a must feature, as we call "Plug and Play". Devices have control right to add, modify and delete configuration of WebAccess on cloud side.

- Devices must support these two way for Plug and Play:
  1. Add full configuration: through the way like 2.3.7. Usually, it is used in scenario as the following:
     - connect to the group_id at first time
     - device configuration was deleted by cloud side
  2. Partial update or delete configuration: through the way like 2.3.8 and 2.3.9.
     Devices should keep a up-to-date configuration profile, and update or delete configuration based on this profile.

- Device replacement scenario:
  Assume A is broken device, and B is new one.
  1. Find A and ready to replace with B.
  2. Set device ID of A to B
  3. Let B connect to cloud and get back up-to-date configuration profile through the way at 2.3.6. B should setup well according to this profile.
  4. Start to publish real-time data through way at 2.1.1 when B all gets ready.
2.4 Command Format

Devices should subscribe these two topics to receive commands from cloud:

\[\text{iot-2/evt/wacmd/fmt/group_id}\]
\[\text{iot-2/evt/wacmd/fmt/group_id/dev_id}\]

2.4.1 Write Value

Write values from cloud to the device

\[\text{iot-2/evt/wacmd/fmt/group_id/dev_id}\]

Format:

```json
{
    "Cmd": "WV",
    "Val": {
        "unique string1": number,
        "unique string2": "string"
        "unique string3": {
            "0": number,
            "1": number,
            "2": number,
            ...
        }
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string1~3</td>
<td>21</td>
<td>Tag ID, must be unique under same WebAccess project and can not be &quot;(&quot; , &quot;&amp;&quot; , &quot;,&quot; , &quot;:&quot; , &quot;.&quot; , &quot;%&quot; , &quot;;&quot; , &quot;:&quot; and space. (Tag ID is mapping to WebAccess Tag Name and Tag Address)</td>
</tr>
</tbody>
</table>
Example:

```json
{
    "d": {
        "Cmd": "WV",
        "Val": {
            "W4010-28A7_Fz1Volt": 1213.48,
            "W4010-28A7_Fz1Temp": 63.81
        }
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```
2.4.2 Write Config

Write configuration from cloud to the device

\texttt{iot-2/evt/wacmd/fmt/group\_id/dev\_id}

Format:

```
{
    "Cmd": "WC",
    "UTg": {
    },
    "DTg": {
    },
    "Del": 1
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTg</td>
<td></td>
<td>Add or Update Tags</td>
</tr>
<tr>
<td>DTg</td>
<td></td>
<td>Delete Tags</td>
</tr>
<tr>
<td>Del</td>
<td>1</td>
<td>Delete All Tags</td>
</tr>
</tbody>
</table>
Example: (modify tag span high and span low, and delete tag)

```json
{
    "d": {
        "Cmd": "WC",
        "UTg": {
            "W4010-28A7_Fz2Volt": {
                "SH": 2000,
                "SL": 200
            },
            "W4010-28A7_Fz2Temp": {
                "SH": 150,
                "SL": -20
            }
        },
        "DTg": {
            "W4010-28A7_Fz1Volt": 1,
            "W4010-28A7_Fz1Temp": 1
        }
    },
    "ts": "2015-04-17T11:58:07+08:00"
}
```
2.4.3 Data On

Indicate devices to start publishing real-time data

\[
\text{iot-2/evt/wacmd/fmt/group_id} \\
\text{iot-2/evt/wacmd/fmt/group_id/dev_id}
\]

Format:

```
{
  "Cmd": "DOn"
}
```

Property:

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOn</td>
<td>start sending real-time data. publish all tag value at first time, then just publish values with changed.</td>
<td></td>
</tr>
</tbody>
</table>

Example:

```
{
  "d": {
    "Cmd": "DOn"
  },
  "ts": "2015-04-17T13:33:29+08:00"
}
```
2.4.4 Data Off

Indicate devices to stop publishing real-time data

```
iot-2/evt/wacmd/fmt/group_id
iot-2/evt/wacmd/fmt/group_id/dev_id
```

Format:

```
{
  "Cmd": "DOf"
}
```

Property:

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOf</td>
<td></td>
<td>stop sending real-time data.</td>
</tr>
</tbody>
</table>

Example:

```
{
  "d": {
    "Cmd": "DOf"
  },
  "ts": "2015-04-17T13:33:29+08:00"
}
```
2.4.5 Backup-Configuration

Indicate the device to backup the current device configuration file to cloud

iot-2/evt/wacmd/fmt/group_id/dev_id

Format:

```
{
    "Cmd": "BkC"
}
```

Property:

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BkC</td>
<td></td>
<td>backup device configuration file</td>
</tr>
</tbody>
</table>

Example:

```
{
    "d": {
        "Cmd": "BkC"
    },
    "ts": "2015-04-17T13:33:29+08:00"
}
```
2.4.6 Restore Configuration

Indicate the device to restore the current device configuration file from cloud
[iot-2/evt/wacmd/fmt/group_id/dev_id]

Format:

```
{
   "Cmd": "RsC",
}
```

Property:

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RsC</td>
<td></td>
<td>resotre device configuration file</td>
</tr>
</tbody>
</table>

Example:

```
{
   "d": {
      "Cmd": "RsC",
   },
   "ts": "2015-04-17T13:33:29+08:00"
}
```
**2.4.7 Firmware Upgrade**

Indicate the device to upgrade firmware from cloud

```
iot-2/evt/wacmd/fmt/group_id/dev_id
```

**Format:**

```json
{
  "Cmd": "FwU"
}
```

**Property:**

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FwU</td>
<td></td>
<td>命令設備更新韌體</td>
</tr>
</tbody>
</table>

**Example:**

```json
{
  "d": {
    "Cmd": "FwU",
    "URL": "http://wavm.cloudapp.net/firmware/WISE4010.bin"
  },
  "ts": "2015-04-17T13:33:29+08:00"
}
```
2.4.8 File Download

Receive download file from cloud

```
iot-2/evt/wacmd/fmt/group_id/dev_id
```

Format:

```
{
    "Cmd": "Fdl",
    "Name": "string",
    "Type": number,
    "Size": number,
    "Bin": binary
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fdl</td>
<td></td>
<td>Files Get from Cloud</td>
</tr>
<tr>
<td>Name</td>
<td>256</td>
<td>File Name</td>
</tr>
<tr>
<td>Type</td>
<td>0 ~ 2</td>
<td>0: ASCII</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: UTF-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Binary</td>
</tr>
<tr>
<td>Size</td>
<td>&lt;=256MB</td>
<td>File Byte Size (Up to 4GB)</td>
</tr>
<tr>
<td>Bin</td>
<td></td>
<td>File Content</td>
</tr>
</tbody>
</table>
Example:

```json
{
   "d": {
      "Cmd": "Fdl",
      "File": {
         "Name": "image.bin",
         "Type": 2,
         "Size": 3576,
         "Bin": "content…"
      }
   },
   "ts": "2015-04-17T13:33:29+08:00"
}
```

(reference request command "FGet" at [2.2.2])
2.4.9 Time Synchronization

Receive time synchronization from cloud

```
iot-2/evt/wacmd/fmt/group_id/dev_id
```

Format:

```
{
   "Cmd": "TSyn",
   "UTC": number
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSyn</td>
<td>Time Synchronization</td>
<td></td>
</tr>
<tr>
<td>UTC</td>
<td>UTC Time from Cloud</td>
<td></td>
</tr>
</tbody>
</table>

Example:

```
{
   "d": {
      "Cmd": "TSyn",
      "UTC": 1460129890
   },
   "ts": "2015-04-17T13:33:29+08:00"
}
```

(reference request command "TSyn" at 2.5.5)
### 2.4.10 Command List

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WV</td>
<td>Val</td>
<td>Write Value</td>
</tr>
<tr>
<td>WC</td>
<td>Cfg</td>
<td>Write Config</td>
</tr>
<tr>
<td>DOn</td>
<td></td>
<td>Data On</td>
</tr>
<tr>
<td>DOf</td>
<td></td>
<td>Data Off</td>
</tr>
<tr>
<td>BkC</td>
<td></td>
<td>Backup Config</td>
</tr>
<tr>
<td>RsC</td>
<td></td>
<td>Restore Config</td>
</tr>
<tr>
<td>FwU</td>
<td></td>
<td>Firmware Upgrade</td>
</tr>
<tr>
<td>Fdl</td>
<td>File</td>
<td>File Download</td>
</tr>
<tr>
<td>TSyn</td>
<td>UTC</td>
<td>Time Synchronization</td>
</tr>
</tbody>
</table>
2.5 Connection Format

Devices publish connection status to this topic:

```
iot-2/evt/waconn/fmt/group_id
```

2.5.1 Connect

Notify cloud side after connecting successfully.

Format:

```
{
  "unique string": {
    "Con": 1
  }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same <code>group_id</code>)</td>
</tr>
<tr>
<td>Con</td>
<td>1</td>
<td>Connect</td>
</tr>
</tbody>
</table>

Example:

```
{
  "d": {
    "WISE4010-7F28A7": {
      "Con": 1
    }
  },
  "ts": "2015-04-17T10:19:51+08:00"
}
```
2.5.2 Disconnect

Notify cloud side after disconnecting successfully.

Format:

```
{
    "unique string": {
        "DsC": 1
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td>DsC</td>
<td>1</td>
<td>Disconnect</td>
</tr>
</tbody>
</table>

Example:

```
{
    "d": {
        "WISE4010-7F28A7": {
            "DsC": 1
        }
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```
2.5.3 Unexpected Disconnect

Implement with MQTT Last Will mechanism to notify cloud side after unexpected disconnection happening.

Format:

```json
{
    "unique string": {
        "UeD": 1
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td>UeD</td>
<td>1</td>
<td>Unexpected Disconnect</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "d": {
        "WISE4010-7F28A7": {
            "UeD": 1
        }
    },
    "ts": "2015-04-17T19:51+08:00"
}
```
2.5.4 Heartbeat

Notify cloud side that the device is alive.

Format:

```
{
   "unique string": {
      "Hbt": 1
   }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td>Hbt</td>
<td>1</td>
<td>Heartbeat for Keep Alive</td>
</tr>
</tbody>
</table>

Example:

```
{
   "d": {
      "WISE4010-7F28A7": {
         "Hbt": 1
      }
   },
   "ts": "2015-04-17T10:19:51+08:00"
}
```
2.5.5 Time Synchronization

Synchronize time from cloud side

Format:

```json
{
    "unique string": {
        "TSyn": 1
    }
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unique string</td>
<td>31</td>
<td>Device ID (unique under same group_id)</td>
</tr>
<tr>
<td>TSyn</td>
<td>1</td>
<td>Time Synchronization</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "d": {
        "WISE4010-7F28A7": {
            "TSyn": 1
        }
    },
    "ts": "2015-04-17T10:51:08:00"
}
```

(reference response command "UTC" at 2.4.9)
2.6 Action Format

Devices publish action to this topic for cloud:

```plaintext
iot-2/evt/waactc/fmt/group_id/dev_id
```

Devices should subscribe these two topics to receive action from cloud:

```plaintext
iot-2/evt/waactd/fmt/group_id
iot-2/evt/waactd/fmt/group_id/dev_id
```

Devices should log these action to help for problem tracking and analyzing.

2.6.1 Connect

Receive the action after connected to cloud successfully (2.5.1).

```plaintext
iot-2/evt/waactd/fmt/group_id/dev_id ("Con": 1)
iot-2/evt/waactd/fmt/group_id ("Con": 2)
```

Format:

```json
{
    "Con": 1
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Con</td>
<td>1</td>
<td>Device Connect Successfully</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Cloud Agent Connect Successfully</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "d": {
        "Con": 1
    },
    "ts": "2015-04-17T10:51+08:00"
}
```
2.6.2 Unexpected Disconnect

Receive the action after cloud agent is offline unexpectedly.

```
iot-2(evt/\waactd/fmt/group_id)
```

Format:

```
{
    "UeD": 1
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UeD</td>
<td>1</td>
<td>Cloud Agent is Offline Unexpectedly</td>
</tr>
</tbody>
</table>

Example:

```
{
    "d": {
        "UeD": 1
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```
2.6.3 Config

Receive the action after updated config to cloud successfully (2.5.1).

\texttt{iot-2(evt/\text{waactd}/fmt/group\_id/dev\_id)}

Notify cloud after updated config from cloud successfully (2.4.2)

\texttt{iot-2(evt/\text{waacta}/fmt/group\_id/dev\_id)}

Format:

\begin{verbatim}
{
    "Cfg": 1
}
\end{verbatim}

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cfg</td>
<td>1</td>
<td>Updated Config Successfully</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Failed to Updated Config</td>
</tr>
</tbody>
</table>

Example:

\begin{verbatim}
{
    "d": {
        "Cfg": 1
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
\end{verbatim}
2.6.4 File

Receive the action after uploaded file handling to cloud successfully (2.2.1).

iot-2(evt/waactd/fmt/group_id/dev_id)

Notify cloud after downloaded file handling from cloud successfully (2.4.8)

iot-2(evt/waacte/fmt/group_id/dev_id)

Format:

```json
{
    "File": 1
}
```

Property:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>1</td>
<td>File Handling is Successfully</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Failed to Handle File</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "d": {
        "File": 1
    },
    "ts": "2015-04-17T10:19:51+08:00"
}
```