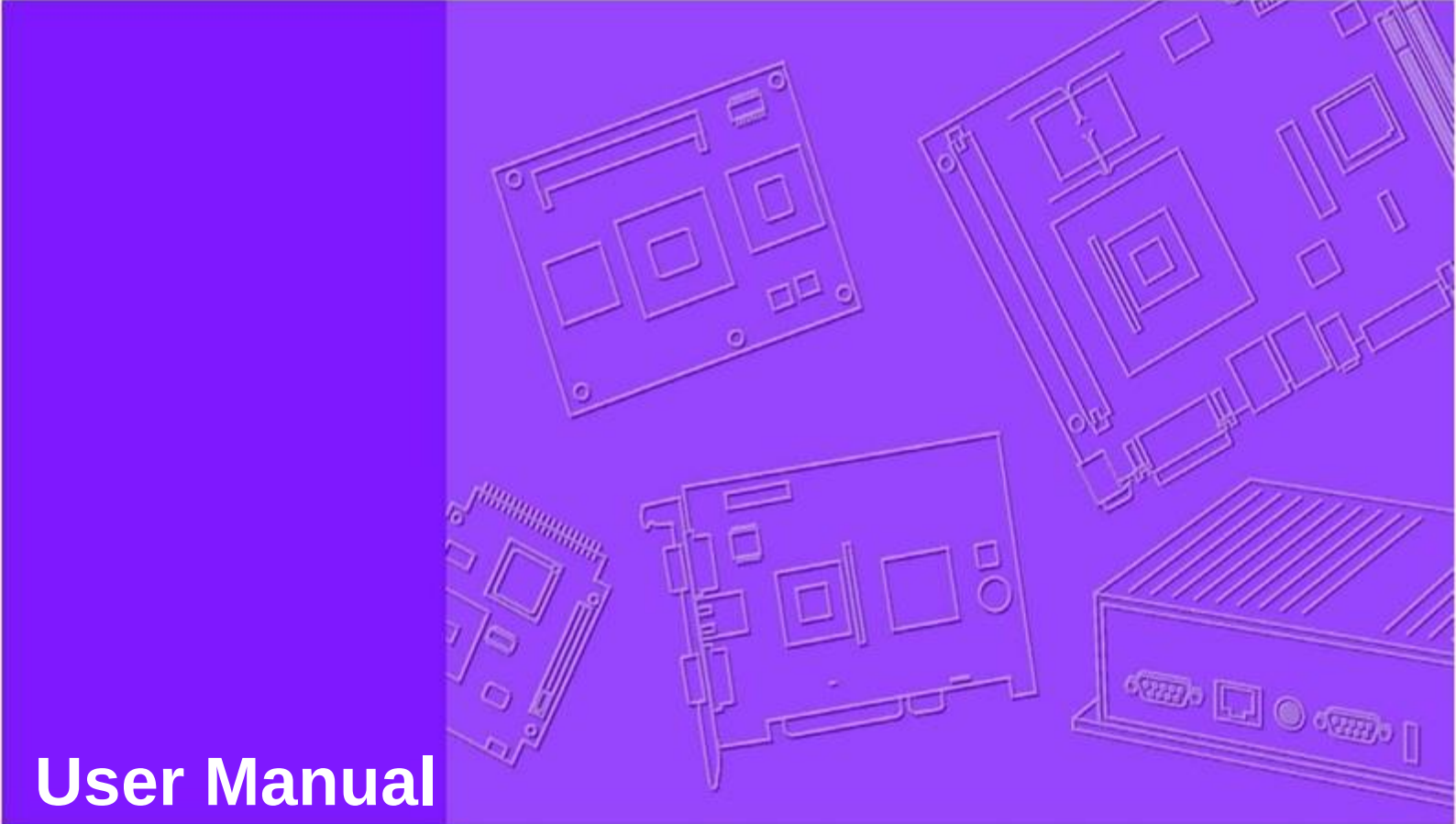




User Manual

WISE-PaaS/RMM 3.1

Wireless IoT Sensing Embedded
Remote Monitoring and Management
Node-RED Plug-ins User Manual

ADVANTECH

Enabling an Intelligent Planet

Change Log:

Date	Version	Description / Major change
2015/08/07	V1.0	Tung Yi, create draft document
2015/08/28	V1.1	Rison Yeh, revise format
2015/08/31	V1.2	Tung Yi, Overview, Category table, flow example, reference
2015/09/10	V1.3	Tung Yi, OAuth support
2015/12/22	V1.4	Rison Yeh, reviewed and revised

Table of Content

1. Overview	4
2. Node Red Plug-in Nodes	5
2.1 General	5
ServerSetting	5
DeviceGet	7
SensorGet	8
FileDownload.....	8
2.2 DeviceQuery	9
SWData	9
HWDData	9
HDDData	9
NetworkData	10
GWSensor	10
SUSIControl.....	11
HDDMonitor	11
ProcessMonitor	11
NetMonitor.....	12
2.3 DeviceCtrl	12
PowerCtrl.....	12
ProtectCtrl	13
AcronisCtrl	14
ScreenShot.....	14
SUSICtrlSet.....	14
2.4 Event.....	15
EmailSend	15
SMSSend.....	16
LongPolling	17
WSRetrieve	17
WSServer	17
2.5 DBCtrl	18
DBCreate.....	18
DBInsert.....	18
DBQuery	19
DBDelete.....	20
DBUpdate	21
3. Installation	23

3.1	Install Node.js	23
3.2	Install Node Red	26
3.3	Install node-red for RMM	27
4.	Flow Control Example	27
5.	Reference url	27

1. Overview

Provide a collection of Node-RED nodes to access RMM services by using standard RESTful APIs. User can use those nodes to combine RMM features for specific application integrating and data flow controlling.

About device query, user can get software/hardware monitoring status, device list, gateway sensor list, and etc. About device control, user can control device power on/off, switch McAfee protection, start Acronis backup and recovery and screen shot. About event manipulation, user can send Email and SMS messages, as well as receive the all events from RMM server when event occurs.

2. Node Red Plug-in Nodes

Category	Description	Nodes
General	The settings of IP address, port number, login username/password of RMM server. Retrieval of device and sensor information.	1. ServerSetting 2. DeviceGet 3. SensorGet 4. FileDownload
DeviceQuery	Device status and information querying, include software/hardware monitoring status, network status, hard disk status and etc.	1. SWData 2. HWDData 3. HDDData 4. NetworkData 5. GWSensor 6. SUSIControl 7. HDDMonitor 8. ProcessMonitor 9. NetMonitor
DeviceCtrl	Remote control of RMM managed devices.	1. PowerCtrl 2. ProtectCtrl 3. AcronisCtrl 4. ScreenShot 5. SUSICtrlSet
Event	Manipulation of message notifying and event receiving from RMM server through long polling or websocket.	1. EmailSend 2. SMSSend 3. LongPolling 4. WSRetrieve 5. WSServer
DBCtrl	Transparent control of SQL database (PostgreSQL) in RMM server.	1. DBCCreate 2. DBInsert 3. DBQuery 4. DBDelete 5. DBUpdate

2.1 General

- **ServerSetting**

Description:

Set url, port, username and password for RMM server. The node support basic mode and OAuth mode. User can switch basic mode and OAuth mode.

Basic mode will request server address, server port, login name and login password.

The screenshot shows a dialog box titled "Edit ServerSetting node". It contains the following fields: "Type" is a dropdown menu set to "Basic"; "Name" is a text field with "Node Name"; "URL" consists of two text fields, "localhost" and "8080"; "Username" is a text field with "admin"; and "Password" is a text field with masked characters (dots). At the bottom right are "Ok" and "Cancel" buttons.

OAuth mode is base on **Authorization Code Grant Flow** of OAuth 2.0. The node will request server address and server port. User need to get credential by clicking red-box button. The node will show "**Click here to authenticate with RMM**" button. If user clicks it and then the node will redirect to RMM credential web page. User can get credential by login RMM credential web page. If login success, the node will save your credential and use it.

The screenshot shows a dialog box titled "Edit ServerSetting node". It contains the following fields: "Type" is a dropdown menu set to "OAuth"; "SA ID" is a dropdown menu with "Add new credential..." and a red box around a pencil icon; "Name" is a text field with "Node Name"; and "URL" consists of two text fields, "172.22.12.195" and "8080". At the bottom right are "Ok" and "Cancel" buttons.

Add new credential config node

Click here to authenticate with SUSIAccess

AddCancel


Remote Manage Your Intelligent Systems Anytime, Anywhere, on Any Devices

Do you want to authorize your account ?

User Name

Password

AuthorizeCancel

Copyright© 2014 Advantech Co., Ltd. All Rights Reserved



Edit ServerSetting node

* Type
OAuth

SA ID
18215.ffe7deb

Name
Node Name

URL
localhost8080

OkCancel

Input:

A timer or button triggered events.

Output:

Message contains properties of *msg.url*, *msg.port* and *msg.encodestr*.

● DeviceGet

Description:

Retrieve device information list from RMM server.

Input:

1. "ServerSetting" node.
2. Properties should be set (msg.url, msg.port, msg.username, msg.pwd) in an incoming message.

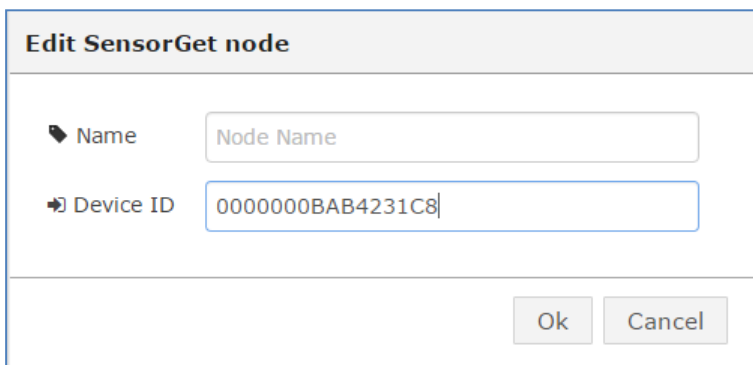
Output:

Device information list with JSON string format.

● **SensorGet**

Description:

Retrieve sensor id list for specific agent ID.



The dialog box is titled "Edit SensorGet node". It contains two input fields: "Name" with the value "Node Name" and "Device ID" with the value "0000000BAB4231C8". At the bottom right, there are "Ok" and "Cancel" buttons.

Input:

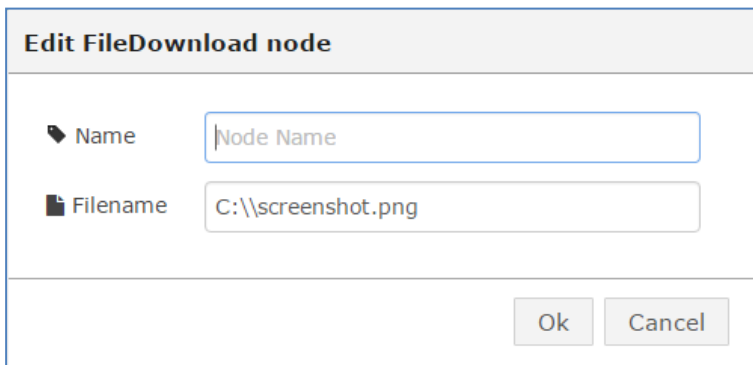
ServerSetting Node or set msg.url, msg.port, msg.username, msg.pwd.

Output:

Sensor information list with JSON string format.

● **FileDownload**

Description: *Download a url link and save to specified path.*



The dialog box is titled "Edit FileDownload node". It contains two input fields: "Name" with the value "Node Name" and "Filename" with the value "C:\\screenshot.png". At the bottom right, there are "Ok" and "Cancel" buttons.

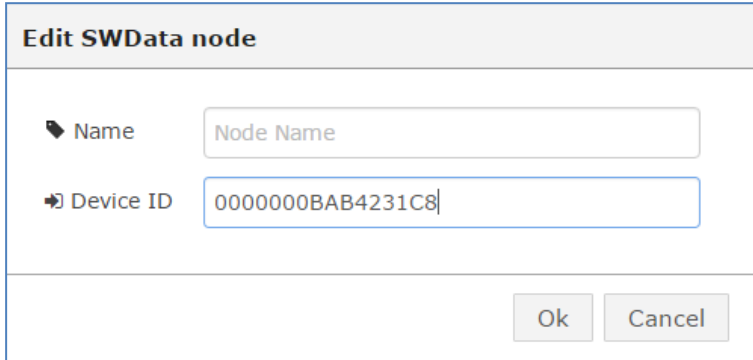
Input: *msg.downloadurl.*

2.2 DeviceQuery

- **SWData**

Description:

Retrieve software and operation system information for specific agent ID.



The dialog box is titled "Edit SWData node". It contains two input fields: "Name" with the placeholder text "Node Name" and "Device ID" with the value "0000000BAB4231C8". At the bottom right, there are "Ok" and "Cancel" buttons.

Input:

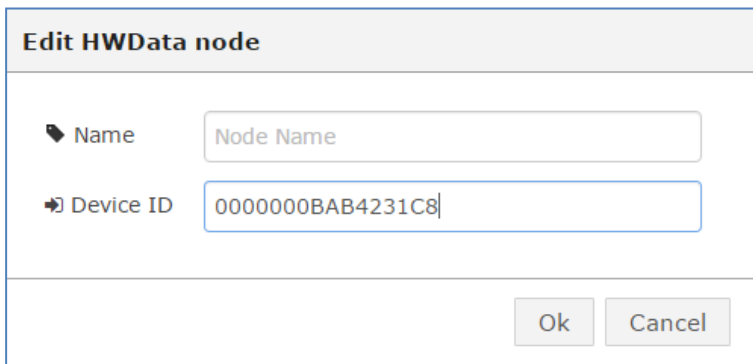
ServerSetting Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output:

Process Monitor information list.

- **HWData**

Description: *Retrieve hardware monitor information of values for specific agent ID.*



The dialog box is titled "Edit HWData node". It contains two input fields: "Name" with the placeholder text "Node Name" and "Device ID" with the value "0000000BAB4231C8". At the bottom right, there are "Ok" and "Cancel" buttons.

Input: **ServerSetting** Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *SUSIoT Information list.*

- **HDDData**

Description: *Retrieve hard disk information for specific agent ID.*

Edit HDDData node

Name

Node Name

Device ID

0000000BAB4231C8

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *HDDMonitor* Information list.

● NetworkData

Description: Retrieve network information for specific agent ID.

Edit NetworkData node

Name

Node Name

Device ID

0000000BAB4231C8

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *NetMonitor* Information list.

● GWSensor

Description: Retrieve the IoTGW sensor reported data from database, depend on sensorId existed.

Edit GWSensor node

Name

Node Name

Device ID

0000000BAB4231C8

Ok

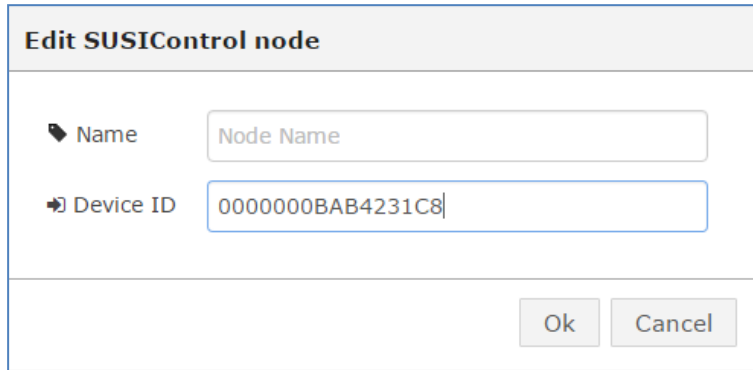
Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *Gateway sensor* Information list.

- **SUSIControl**

Description: *Retreive the SUSIControl reported data from database, depend on sensorId existed.*



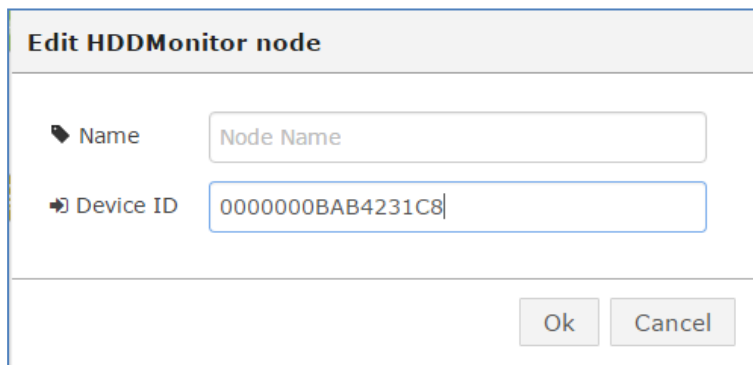
The dialog box titled "Edit SUSIControl node" contains two input fields. The first field, labeled "Name" with a folder icon, contains the text "Node Name". The second field, labeled "Device ID" with a key icon, contains the text "0000000BAB4231C8". At the bottom right, there are two buttons: "Ok" and "Cancel".

Input: **ServerSetting** Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: *SUSIIoT Information list.*

- **HDDMonitor**

Description: *Retreive the HDDMonitor reported data from database, depend on sensorId existed.*



The dialog box titled "Edit HDDMonitor node" contains two input fields. The first field, labeled "Name" with a folder icon, contains the text "Node Name". The second field, labeled "Device ID" with a key icon, contains the text "0000000BAB4231C8". At the bottom right, there are two buttons: "Ok" and "Cancel".

Input: **ServerSetting** Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: *HDDMonitor Information list.*

- **ProcessMonitor**

Description: *Retreive the ProcessMonitor reported data from database, depend on sensorId existed.*

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *Process Monitor information list*.

- **NetMonitor**

Description: *Retreive the NetMonitor reported data from database, depend on sensorId existed.*

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *NetMonitor Information list*.

2.3 DeviceCtrl

- **PowerCtrl**

Description: *Set device to power on, power off, sleep, hibernate, reboot.*

Edit PowerCtrl node

Name

Node Name

Device ID

0000000BAB4231C8

Method

Reboot

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: { "result" : true } or error code description

● ProtectCtrl

Description: Set device to install/update McAfee, protect, unprotect, activate.

Edit ProtectCtrl node

Name

Node Name

Device ID

0000000BAB4231C8

Method

Activate

Company

Company Name

SN

Serial Number

Ok

Cancel

Edit ProtectCtrl node

Name

Node Name

Device ID

0000000BAB4231C8

Method

Install

Activate

False

Ok

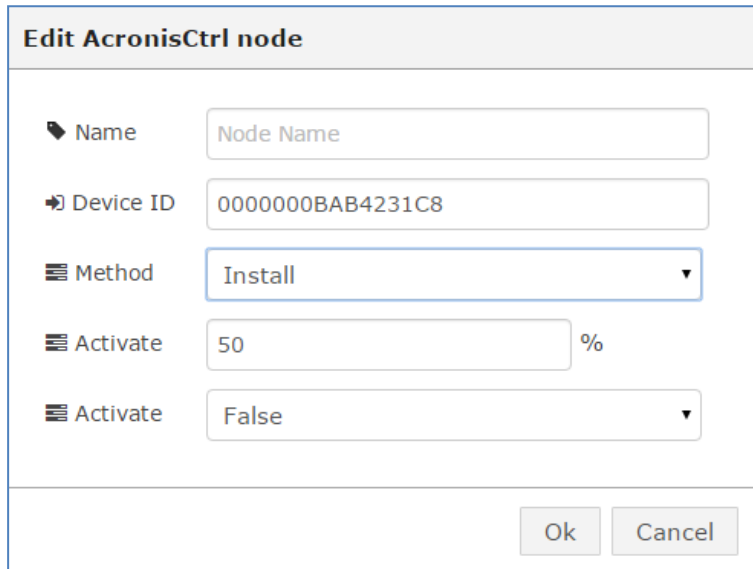
Cancel

Input: *ServerSetting* Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: { "result" : true } or error code description

- **AcronisCtrl**

Description: Set device to install/update Acronis, backup, recovery, activate.



The dialog box is titled "Edit AcronisCtrl node". It contains the following fields:

- Name:** A text input field with the value "Node Name".
- Device ID:** A text input field with the value "0000000BAB4231C8".
- Method:** A dropdown menu with "Install" selected.
- Activate:** A text input field with the value "50" and a percentage symbol (%) to its right.
- Activate:** A dropdown menu with "False" selected.

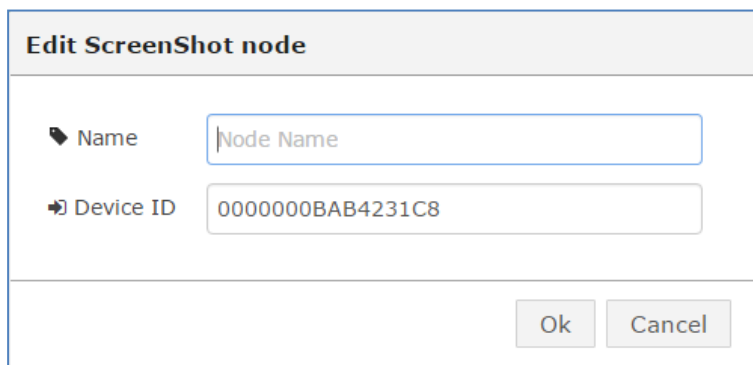
At the bottom right, there are two buttons: "Ok" and "Cancel".

Input: *ServerSetting* Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: { "result" : true } or error code description

- **ScreenShot**

Description: Retrieve current screenshot for specific device.



The dialog box is titled "Edit ScreenShot node". It contains the following fields:

- Name:** A text input field with the value "Node Name".
- Device ID:** A text input field with the value "0000000BAB4231C8".

At the bottom right, there are two buttons: "Ok" and "Cancel".

Input: *ServerSetting* Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: Get a relative path for specific device.

- **SUSICtrlSet**

Description: Set SUSIoT value to specific device.

Edit SUSICtrlSet node

Name
Node Name

URL
172.22.13.25
8080

Username
admin

Password
.....

Device ID
0000000BAB4231C8

Sensor ID
/GPIO/GPIO07/Dir

Set Value
value
Set Value

Ok
Cancel

Input: *ServerSetting* Node or set msg.url, msg.port, msg.username, msg.pwd.

Output: { "result" : true } or error code description

2.4 Event

- **EmailSend**

Description: Send the Email by using RMM 3.1 server.

Edit EmailSend node

Name
Node Name

Server
mail.advantech.com.tw

Port
25
SSL

Account
tung.yi

Password
.....

Sender
tung.yi@advantech.com.tw

Subject
Subject

Recipient
tung.yi@advantech.com.tw

Test send

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: { "result" : true } or error code description

● SMSSend

Description: Send the SMS by using RMM 3.1 server.

Edit SMSSend node

Name
Node Name

API ID
clickatell id

Account
clickatell account

Password
.....

Phone
886910526771

test send

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: { "result" : true } or error code description

● LongPolling

Description: Retrieve event from database, if event be generated and newer event id larger than "lasteventid". It will be kept 60s on this request without event be generated.

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: Get the event of system, operator and device when the event is occurred.

● WSRetrieve

Description: Retrieve event from database by using websocket.

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: Get the event of system, operator and device when the event is occurred.

● WSServer

Description: Provide a websocket server for port 9876, the node will broadcast when *msg.payload* entry.

Input: *msg.payload*.

2.5 DBCtrl

- **DBCreate**

Description: *Create a table with the name, length and type of column under RMM 3.1 postgre database.*

Edit DBCreate node

 Name

 TableName

 Field Name

 Field Type

 Length

 Allow NULL

 Add

	name	type	length	allownull
	aaaa	character	256	true
	bbbb	timestamp		true
	cccc	double		true

Ok

Cancel

Input: *ServerSetting Node or set msg.url, msg.port, msg.username, msg.pwd.*

Output: *{ "result" : true } or error code description.*

- **DBInsert**

Description: *Insert data into an existed table of RMM 3.1 postgre database.*

Edit DBInsert node

Name
Node Name

URL
172.22.13.25
8080

Username
admin

Password
.....

TableName
mytable

+ Get Table Columns

column name	value
aaaa(char)	123
bbbb(ts)	2015/08/06
cccc(double)	345

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: { "result" : true } or error code description.

- **DBQuery**

Description: Query data from an existed table of RMM 3.1 postgre database.

19

Edit DBQuery node

Name
Node Name

URL
172.22.13.25
8080

Username
admin

Password
.....

TableName
mytable

+ Get Columns
Fields

CheckBox	selectFields
☒	aaaa
☒	bbbb
☒	cccc

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: *Query data from specific table.*

DBDelete

Description: *Delete data from an existed table of RMM 3.1 postgres database.*

20

Edit DBDelete node

Name
Node Name

URL
172.22.13.25
8080

Username
admin

Password
.....

TableName
mytable

+ Get Columns
Conditions

	FieldName	Operator	Value
✕	aaaa	=	555

CCCC

=

777

+ Add Condition

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

Output: Delete data from specific table.

● DBUpdate

Description: Change data from a existed table of RMM 3.1 postgres database.

21

Edit DBUpdate node

Name
Node Name

URL
172.22.13.25
8080

Username
admin

Password
.....

TableName
mytable

+ Get Columns
Fields

	Field	Value
x	aaaa	newvalue

aaaa
newvalue

+ Set Fields

Ok

Cancel

Input: *ServerSetting* Node or set *msg.url*, *msg.port*, *msg.username*, *msg.pwd*.

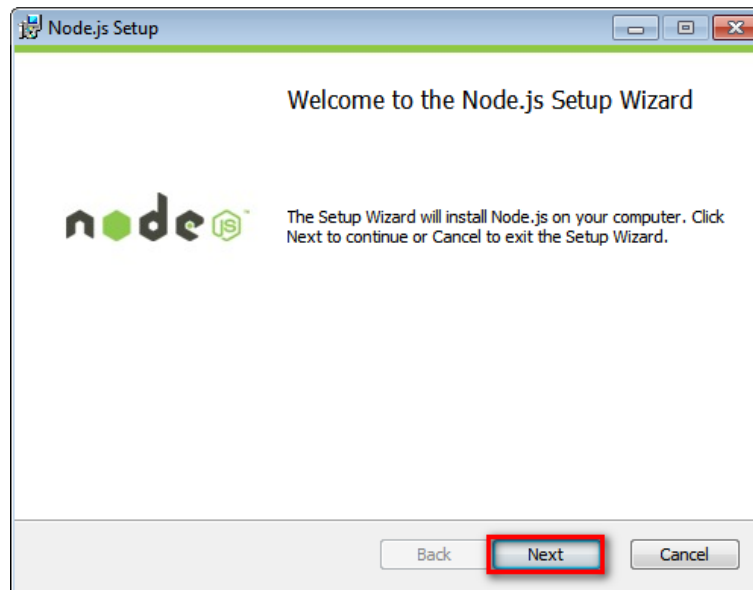
Output: Update data from specific table.

22

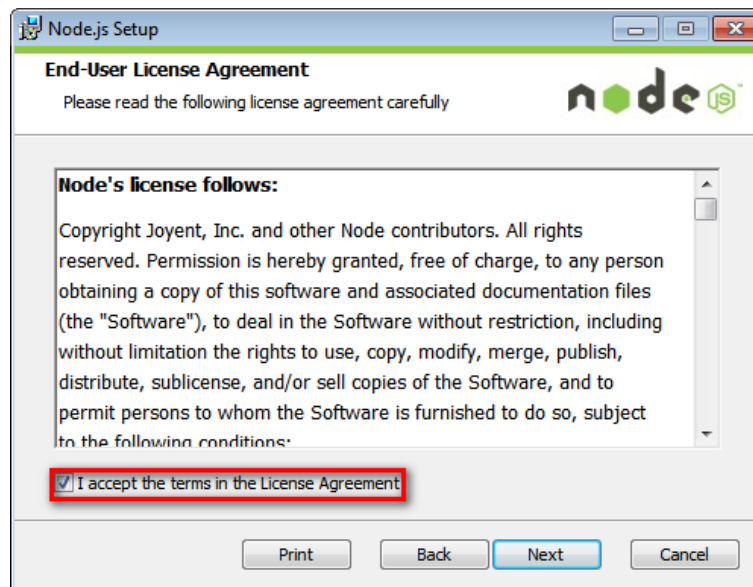
3. Installation

3.1 Install Node.js

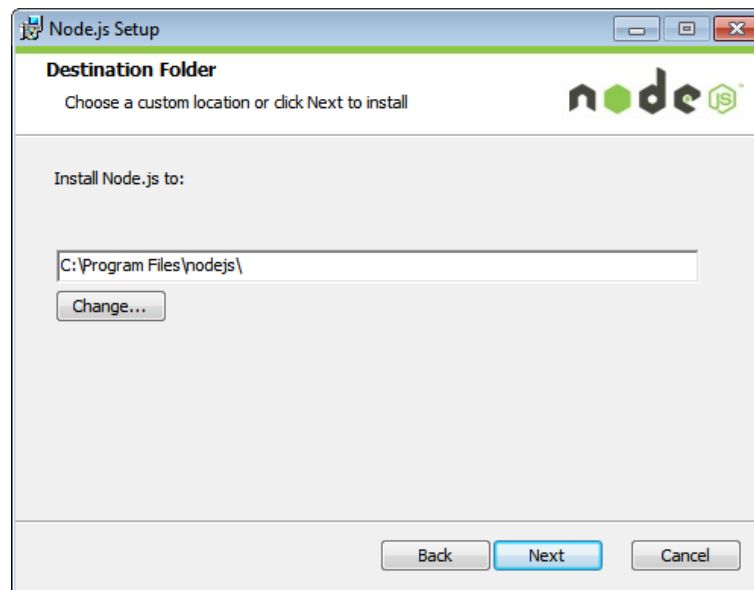
- Step 1: Download installer from <https://nodejs.org>
 - Step 2: Double-click to install **Node.js**
- Click Next to start Setup Wizard



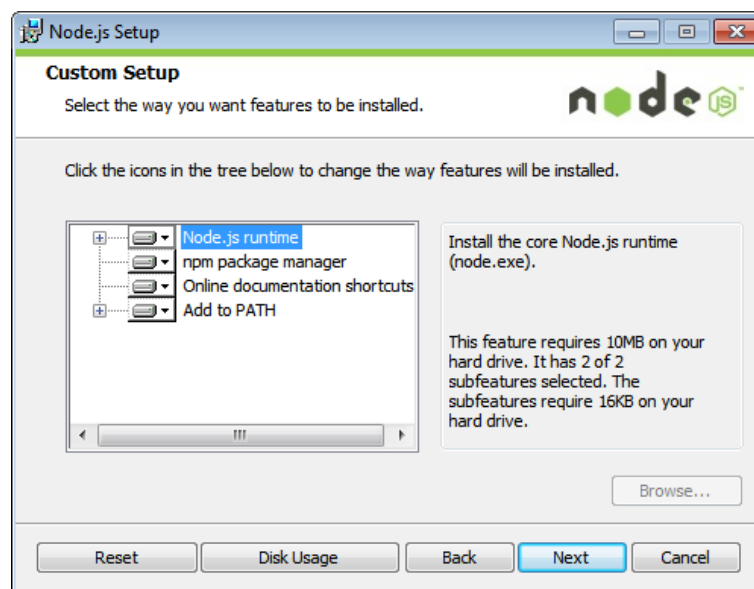
Accept License Agreement



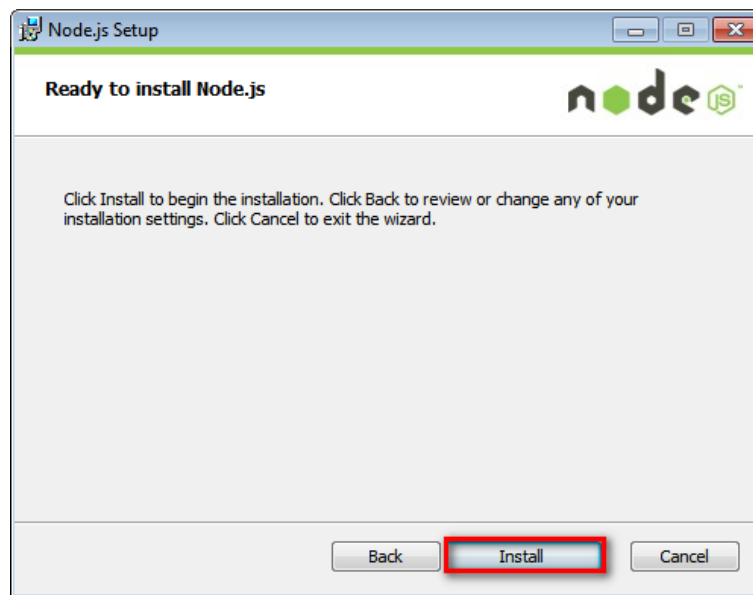
Select installation path



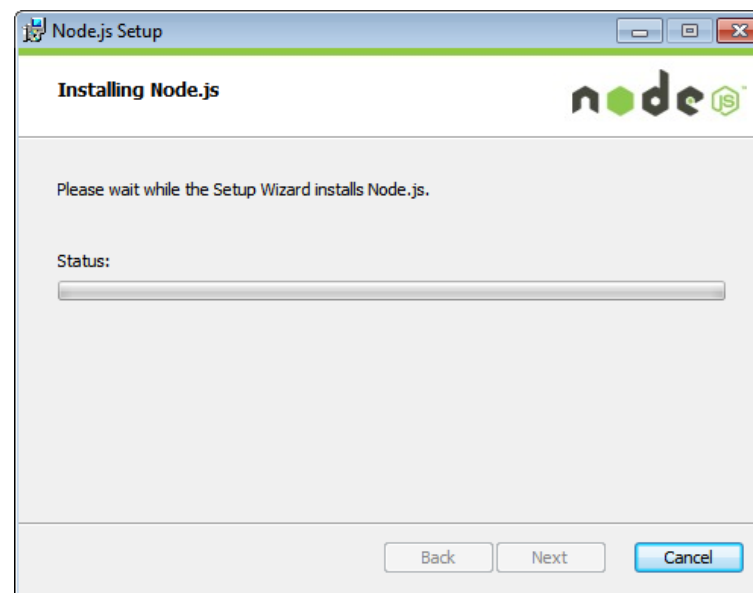
Custom selection



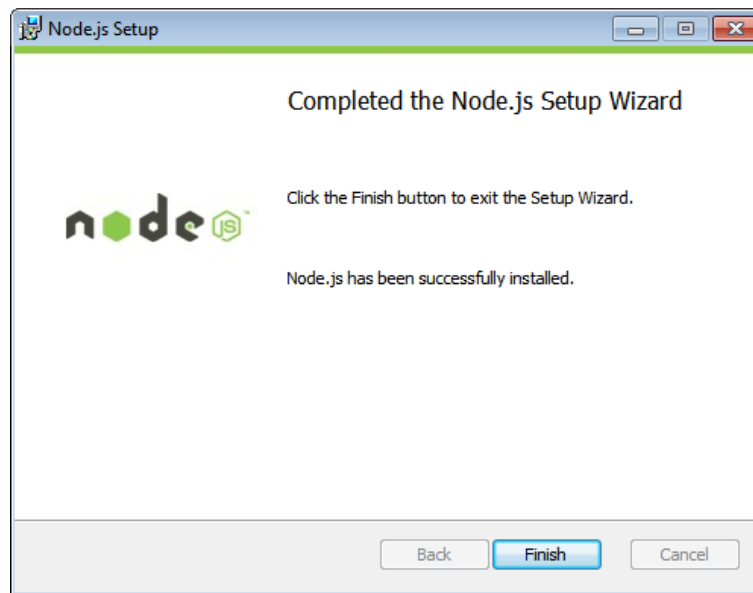
Begin the installation



Installation progress



Installation complete



3.2 Install Node Red

- Open "cmd.exe" and type "**npm install -g node-red**"

```
系统管理员: 命令提示符
d\node_modules\ws\node_modules\bufferutil\build\Release\bufferutil.lib and ob
ject C:\Users\tung.yi\AppData\Roaming\npm\node_modules\node-red\node_modules\
ws\node_modules\bufferutil\build\Release\bufferutil.exp
Generating code
Finished generating code
bufferutil.vcxproj -> C:\Users\tung.yi\AppData\Roaming\npm\node_modules\node-
red\node_modules\ws\node_modules\bufferutil\build\Release\bufferutil.node
> serialport@1.7.4 install C:\Users\tung.yi\AppData\Roaming\npm\node_modules\nod
e-red\node_modules\node-red-node-serialport\node_modules\serialport
> node-pre-gyp install --fallback-to-build
[serialport] Success: "C:\Users\tung.yi\AppData\Roaming\npm\node_modules\node-re
d\node_modules\node-red-node-serialport\node_modules\serialport\build\serialport
.v1.7.4\Release\node-v11-win32-x64\serialport.node" is installed via remote
C:\Users\tung.yi\AppData\Roaming\npm\node-red -> C:\Users\tung.yi\AppData\Roa
ming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\tung.yi\AppData\Roaming\npm\node-red -> C:\Users\tung.yi\AppData\Roamin
g\npm\node_modules\node-red\red.js
node-red@0.11.1 C:\Users\tung.yi\AppData\Roaming\npm\node_modules\node-red
is-utf8@0.2.0
on-headers@1.0.0
node-red-node-rbe@0.1.0
clone@1.0.2
mustache@2.1.2
semver@5.0.1
passport-oauth2-client-password@0.1.2 <passport-strategy@1.0.0>
cors@2.7.1 <vary@1.0.1>
fs.notify@0.0.4 <async@0.1.22, retry@0.6.1>
nopt@3.0.3 <abbrev@1.0.7>
passport-http-bearer@1.0.1 <passport-strategy@1.0.0>
bcryptjs@2.2.0
passport@0.2.2 <pause@0.0.1, passport-strategy@1.0.0>
when@3.7.3
oauth2orize@1.0.1 <uid2@0.0.3, utils-merge@1.0.0, debug@0.7.4>
raw-body@2.1.2 <bytes@2.1.0, unpipe@1.0.0, iconv-lite@0.4.11>
sentiment@0.2.3 <async@0.2.10, lodash@1.3.1>
xml2js@0.4.9 <sax@0.6.1, xmlbuilder@2.6.4>
cron@1.0.9 <moment-timezone@0.3.1>
mqtt@0.3.13 <readable-stream@1.0.33>
follow-redirects@0.0.6 <debug@2.2.0>
i18next@1.10.2 <json5@0.2.0, i18next-client@1.10.2, cookies@0.5.0>
uglify-js@2.4.23 <uglify-to-browserify@1.0.2, async@0.2.10, yargs@3.5.4,
source-map@0.1.34>
cheerio@0.19.0 <entities@1.1.1, dom-serializer@0.1.0, css-select@1.0.0, h
tmlparser2@3.8.3, lodash@3.10.1>
node-red-node-email@0.1.0 <imap@0.8.14, nodemailer@1.3.4>
fs-extra@0.22.1 <jsonfile@2.2.1, graceful-fs@4.1.2, rimraf@2.4.2>
node-red-node-twitter@0.1.1 <twitter-ng@0.6.2, oauth@0.9.13, request@2.57
.0>
node-red-node-feedparser@0.1.0 <feedparser@0.19.2, request@2.57.0>
express@3.20.3 <basic-auth@1.0.0, merge-descriptors@1.0.0, utils-merge@1
.0.0, cookie-signature@1.0.6, escape-html@1.0.1, fresh@0.2.4, cookie@0.1.2, metho
ds@1.1.1, range-parser@1.0.2, vary@1.0.1, content-type@1.0.1, parseurl@1.3.0, co
nvent-disposition@0.5.0, commander@2.6.0, depd@1.0.1, debug@2.2.0, proxy-addr@1
.0.8, etag@1.5.1, mkdirp@0.5.0, send@0.12.3, connect@2.29.2>
bcrypt@0.8.3 <bindings@1.2.1, nan@1.8.4>
ws@0.7.2 <options@0.0.6, ultron@1.0.2, utf-8-validate@1.1.0, bufferutil@1
.1.0>
node-red-node-serialport@0.0.2 <serialport@1.7.4>
C:\Users\tung.yi>
```

3.3 Install node-red for RMM

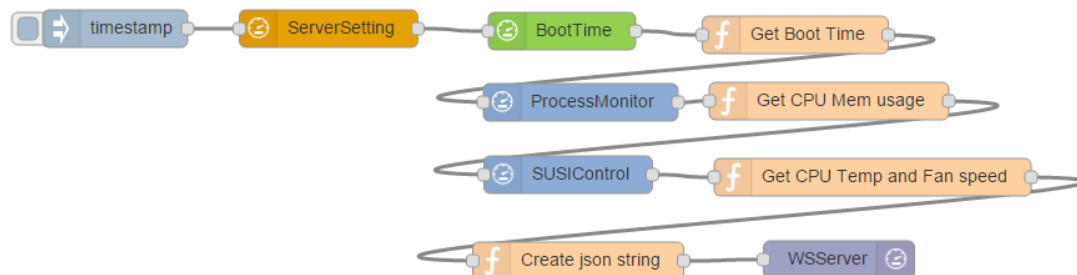
- Set working directory as “users\<user name>”
- Open “cmd.exe” and type “npm i node-red-contrib-rmm3.1”

4. Flow Control Example

Example 1. User wants to take a screen shot for specific device and save the png file into a specific location every morning at 7:00 a.m.



Example 2. User wants to get boot time, software monitoring, fan speed and CPU temperature of a specific device and combine into JSON format. And then send the JSON string to web-socket.



5. Reference url

<https://www.npmjs.com/package/node-red-contrib-rmm3.1>

<https://github.com/ADVANTECH-Corp/node-red-contrib-rmm-3.1>