The JSONNotify Driver

The JSONNotify driver provides a web-based method to push data from the North device, allowing you to receive information from it. Provide a URL, select when and how you want that URL to receive data, and the driver will send it. Available for Commander and ObSys.

This document relates to JSONNotify driver version 2.0

Please read the *Commander Manual* or *ObSys Manual* alongside this document, available from *www.northbt.com*

Contents

Purpose of JSONNotify Driver

The JSONNotify driver provides a web-based method to push data from the North device, allowing you to receive information from it. Provide a URL, select when and how you want that URL to receive data, and the driver will send it.

Use this web service to integrate data collected using North interface technology directly into your own application. The service sends data off-site to your server, bypassing the need for inbound firewall rules or VPNs.

The JSONData driver is also available, providing a web-based API for your application to request and adjust information from the North device.

Values

JSONNotify presents values from the North device's Essential Data and Extra Data. Essential Data contains 640 values on Commander, and 1280 values on ObSys. If necessary, start the Extra Data interface (which requires an interface licence) for an additional 1024 values. Access to these values can be controlled by configuring privilege levels within the driver.

The driver sends these values using one of the following protocols:

- Webhooks – as a value updates in Essential Data or Extra Data, these updated values are sent in near real-time; or all values can be sent periodically in JSON format
- WebDAV – all values are sent periodically in CSV format
- Log data notifications – periodically, all log-enabled values are sent in CSV format

Prerequisites

The Essential Data module should be configured. The JSONNotify driver requires Essential Data v3.0 (build 01/09/2015) or later.

The North device should have a DNS Server address and Gateway address configured.

Detailed Operation

Endpoint Server

When an event occurs, the driver sends an HTTP request to the URL you've specified. If that URL is unavailable or takes too long to respond, the driver will cancel the request and try again a minute later.

The endpoint server must be HTTP 1.1 compatible, and support chunked transfer encoding.

Configure the URL using the Server DNS, Server Port, and Server Resource Name objects. The resource name supports variables; these can be used to dynamically change the URL to include the date, time, or notify of a database revision.

The following variables may be included in the server resource name:

- North device's serial number
- North device label
- Current date
- Current time
- Flag indicating database structure has changed
- Data format – e.g. 'csv' or 'json'

Security

To protect your API endpoint from unauthorised requests, there are two ways that JSONNotify can identify itself with your application:

- HTTP Basic authentication – authorize an individual request for data
- API key – a fixed key included in the URL

The driver supports HTTP URLs only, with a selectable TCP port number. HTTPS is not supported.

Webhooks

Webhooks let you easily add push notifications to your own applications. A Webhook is an HTTP POST callback request in JSON format sent to a URL of your choice in response to an event occurring.

JSONNotify sends a Webhook containing only relevant values when the following event occurs:

- A value changes in Essential Data or Extra Data

JSONNotify sends a Webhook containing all database values when any of the following events occur:

- Timer expires – use to resynchronize all data at midnight for example
- Alarm message received
- Database structure changes

Request Format

The driver sends an HTTP POST request with several parameters, as documented below. Parameters are formatted as JSON objects with UTF-8 character encoding (application/json; charset=utf-8).

The driver will close the connection after sending the request. No response is required.

Body

```
{
   "version": number,
   "id": string,
   "label": string,
   "date": string,
   "revision": number,
```

```
   "obj": [
      {
         "oid": number,
         "type": string,
         "value": various,
         "updated": string,
         "isUnreliable": boolean,
         "isOutRange": boolean,
         "label": string
      }, ...
   ]
}
```

Parameters

Parameter	Value	Description
version	Number	JSONNotify data version – 1
id	String	Serial number of North device
label	String	System label configured in the driver, or if within obj array, label of object (max. 43 chars)
date	String	Current date and time (UTC) in ISO-8601 format
revision	Number	Database revision number. This will increment when an object is added or amended to Essential Data or Extra Data.
obj	Array	Array of values from Essential Data
oid	Number	Object identifier is in the range 1…1280 for Essential Data (depending on platform), and 2001…3024 for Extra Data
type	String	Type of object, one of the following: text, noyes, offon, num, enum, float, datetime, date, times, or profile
value	Number, Boolean, String, or Array	Value of object. The value depends on the object type configured in Essential Data: Float, Num, ENum – Number OffOn, NoYes – Boolean Text – String (max. 32 chars) DateTime – String in ISO-8601 format, '*yyyy-mm-dd*T*hh:mm:ss*' Date – String in ISO-8601 format, '*yyyy-mm-dd*' Times – Array of objects – each containing a start (s) and end (e) time Profile – Array of objects – each containing a time (s) and value (v)
s	String	Start time in 24hr format, 'hh:mm'. Provided with Times and Profile types
e	String	End time in 24hr format, 'hh:mm'. Only provided with times types
v	Number	Profile value. Only provided with profile types
updated	String	Date and time value last updated (local time) in ISO-8601 format, '*yyyy-mm-dd*T*hh:mm:ss*'
isUnreliable	Boolean	If 'true', communications with the sub-system is failing
isOutRange	Boolean	If 'true', value is outside min/max limits
units	String	Optional units for object value (max. 8 chars)
enum	Array	Optional array of objects, each containing an index (i) and label (l) for each value. Only provided with enum types
i	Number	Index value of enumeration. Only provided with enum types.
l	String	Label for enumeration. Only provided with enum types.

Example

In this Webhooks example, JSONNotify sends all the data available as a timer event occurred.

```
POST /websocket/endpoint?id=99999999&apiKey=49B4C0DEA67 HTTP/1.1
Host: test.northbt.com
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8
Connection: close
```

```
{
    "version": 1,
    "id": "99999999",
    "label": "North JSON Notify",
    "date": "2014-01-01T00:00:00Z",
    "revision": 1,
    "obj": [
      {
         "oid": 17,
         "type": "float",
         "value": 12.3,
         "updated": "2013-12-31T23:40:16",
         "isUnreliable": false,
         "isOutRange": false,
         "units": "W",
         "label": "UPS Status - Load power"
      },
      {
         "oid": 18,
         "type": "num",
         "value": 43,
         "updated": "2013-11-16T14:32:28",
         "isUnreliable": true,
         "isOutRange": false,
         "units": "mins",
         "label": "UPS Status - Battery Time Left"
      },
      {
         "oid": 19,
         "type": "enum",
         "value": 1,
         "enum": [{"i": 1, "l": "Healthy"}, {"i": 2, "l": "Replace"}],
         "updated": "2013-12-31T23:40:16",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Battery"
      },
      {
         "oid": 20,
         "type": "noyes",
         "value": false,
         "updated": "2013-12-31T23:40:16",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Test"
      },
      {
         "oid": 21,
         "type": "text",
         "value": "Smart-UPS 450",
         "updated": "2013-12-31T23:32:27",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Model"
      },
      {
         "oid": 22,
         "type": "datetime",
```

```
         "value": "2014-01-01T00:00:00",
         "updated": "2014-01-01T00:00:00",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Date"
      },
      {
         "oid": 24,
         "type": "times",
         "value": [{"s": "00:00", "e": "08:30"}, {"s": "18:00", "e": "24:00"}],
         "updated": "2013-12-31T23:40:17",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Alarm"
      },
      {
         "oid": 28,
         "type": "profile",
         "value": [{"s": "00:00", "v": 18}, {"s": "09:00", "v": 21}],
         "updated": "2013-12-31T23:40:18",
         "isUnreliable": false,
         "isOutRange": false,
         "label": "UPS Status - Setpoint"
      }
   ]
}
```

WebDAV

The WebDAV protocol allows you to manage files on a web server. JSONNotify implements the HTTP PUT request to create a file in CSV format at a URL of your choice.

JSONNotify creates a file containing all database values when any of the following events occur:

- Timer expires
- Alarm message received
- Database structure changes

Request Format

The driver sends an HTTP PUT request containing a comma-separated value (CSV) file, starting with a four-line header followed by the data.

The driver will close the connection after sending the request. No response is required.

Body

```
#id,id
#label,devLabel
#date,date
Label,Value,Units,State,Updated,Type,Params
label,value,units,state,updated,type,params
label,value,units,state,updated,type,params
...
```

Parameters

Parameter	Value	Description
id	String	Serial number of North device
devLabel	String	System label configured in the driver, always in double-quotes
date	String	Current date and time (UTC) in format '*DD/MM/YY hh:mm:ss*'
label	String	Label of object (max. 43 chars), always in double-quotes
value	Number or String	Value of object. The value depends on the object type configured in Essential Data: Float, Num, ENum, OffOn, NoYes – Number Text – String (max. 32 chars), always in double-quotes DateTime – String in format '*DD/MM/YY hh:mm:ss*' Date – String in format, '*DD/MM/YY*' Times – List of on-off times, in the format '*hh:mm-hh:mm,hh:mm-hh:mm*' Profile – List of time-values, in the format '*hh:mm=number,hh:mm=value*'
units	String	Optional units for object value (max. 8 chars)
state	Number	Alarm state: 0=OK, 1=Alarm, 2=Comms fault, 3=Alarm & Comms fault
updated	String	Date and time value last updated (local time) in format '*DD/MM/YY hh:mm:ss*'
type	String	Type of object, one of the following: Text, NoYes, OffOn, Num, ENum, Float, DateTime, Date, Times, or Profile
params	String	Optional list of enumerated strings available for value. Only provided with ENum types, always in double-quotes

Example

In this WebDAV example, JSONNotify sends the data as a CSV file.

```
PUT /webdav/endpoint/file.csv HTTP/1.1
Host: test.northbt.com
Transfer-Encoding: chunked
Content-Type: text/csv; charset=iso-8859-1
Connection: close

#id,99999999
#label,North JSON Notify
#date,01/01/14 00:00:00
Label,Value,Units,State,Updated,Type,Params
"UPS Status - Load power",12.3,W,0,31/12/13 23:40:16,Float,
"UPS Status - Battery Time Left",43,mins,2,16/11/2013 14:32:28,Num,
"UPS Status - Battery",1,,0,31/12/13 23:40:16,ENum,",Healthy,Replace"
"UPS Status - Test",0,,0,31/12/14 23:40:16,NoYes,
"UPS Status - Model","Smart-UPS 450",,0,31/12/13 23:32:27,Text,
"UPS Status - Date",01/01/14 00:00:00,,0, 01/01/14 00:00:00,DateTime,
"UPS Status - Alarm","00:00-08:30,18:00-24:00",,0,31/12/13 23:40:17,Times,
"UPS Status - Setpoint","00:00=18,09:00=21",,0,31/12/13 23:40:18,Profile,
```

Log Data Notifications

The Log Data protocol allows you to receive a list of logged values periodically, for appending to a file. A Log Data notification is an HTTP POST request in CSV format sent to a URL of your choice.

JSONNotify sends a single line containing all log-enabled database values when any of the following events occur:

- Timer expires
- Alarm message received
- Database structure changes

Request Format

The driver sends an HTTP POST request containing a single comma-separated value (CSV) line. Only values from Essential Data, with data log enabled set to 'yes', are included.

When sending the first notification, or when the database structure has changed, a three-line header is included. An option to always send the header is also available.

The driver will close the connection after sending the request. No response is required.

Body

```
#id,id
#label,devLabel
Date,label,label,label,label,label,label...
date,value,value,value,value,value,value...
```

Parameters

Parameter	Value	Description
id	String	Serial number of North device
devLabel	String	System label configured in the driver
label	String	Label of object (max. 43 chars), always in double-quotes
date	String	Current date and time (UTC) in format '*DD/MM/YY hh:mm:ss*'
value	Number	Value of object

Example

In the Log Notification example, JSONNotify initially sends the following request. As this is the first notification since the database structure has changed, the header is included.

```
POST /log/endpoint/line.csv HTTP/1.1
Host: test.northbt.com
Transfer-Encoding: chunked
Content-Type: text/csv; charset=iso-8859-1
Connection: close

#id,99999999
#label,North JSON Notify
Date,"UPS Status - Load power","UPS Status - Battery Time Left"
01/01/14 00:00:00,12.3,43
```

Following this, only the values are sent.

```
POST /log/endpoint/line.csv HTTP/1.1
Host: test.northbt.com
Transfer-Encoding: chunked
Content-Type: text/csv; charset=iso-8859-1
Connection: close

01/01/14 00:05:00,15.6,43
```

Using the Driver

On ObSys and Commander, the JSONNotify driver is pre-installed. Once started, you will need to set up the driver before it can communicate with an HTTP endpoint.

Starting the Interface

💻 To start an interface using the JSONNotify driver, follow these steps:

- → **Start Engineering** your North device using ObSys
- → Navigate to **Configuration, Interfaces,** and set an unused **Interface** to 'JSONNotify' to start the particular interface
- → Navigate to the top-level of your North device and re-scan it

The driver setup object (Mc), labelled **JSON Notify Setup**, should now be available. If this object is not available, check an interface licence is available and the driver is installed.

Setting up the Driver

💻 To set up the driver, follow these steps:

- → Navigate to the **JSON Notify Setup** object (M*c*). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'
- → Set **Notification Type** to the method supported by your HTTP endpoint
- → Navigate to **Destination Server** (N) and set **Server IP/DNS, Server Port** and **Server Resource Name** objects to the URL that will receive information.

Checking Communications

After configuring Essential Data or Extra Data, the driver will automatically send information to the URL.

Use the driver objects **Destination Available** (DS) and **Last Response Code** (LRC) to check if information was received successfully.

Object Specifications

Once an interface is started, one or more extra objects become available within the top-level object of the device. As with all North objects, each of these extra objects may contain sub-objects, (and each of these may contain sub-objects, and so on) - the whole object structure being a multi-layer hierarchy. It is possible to navigate around the objects using the ObSys Engineering Software.

Each object is specified below, along with its sub-objects.

Device Top-Level Objects

When an interface is started using the JSONNotify driver, the objects below become available within the top-level object of the device. For example, if interface 1 is started, then the object reference 'M1' becomes available.

Description	Reference	Type
JSON Notify Setup Set up the JSONNotify driver, started on interface *c* (*c* is the interface number)	M*c*	Fixed Container: On the Commander platform this will be *[CDM v20\JSONNotify v20]* On the ObSys platform this will be *[OSM v20\JSONNotify v20]*
North JSON Notify	S*c*	Fixed Container: *[JSONNotify v20]*

JSON Notify Setup

Object Type: [OSM v20\JSONNotify v20]
Object Type: [CDM v20\JSONNotify v20]

The JSONNotify driver contains the following objects.

Description	Reference	Type
System label Label displayed when scanning the system and sent with request to endpoint server	DL	Obj\Text; Max 20chars; Adjustable
Notification Type Method used to communicate with server. Refer to the beginning of this document	TY	Obj\ENum: 1…3; Adjustable Values: 0=Off, 1=WebDAV, 2=Webhook, 3=Log data
Timer Period Frequency that a Timer event should be triggered, causing all data to be sent to the server. For Webhook notifications, set this to Off or 00:00, as values are sent on change	SP	Obj\ENum: 0…8; Adjustable Values: 0=Off, 1=5mins, 2=15mins, 3=30mins, 4=1hr, 5=3hrs, 6=6hrs, 7=12hrs, 8=00:00
Log Data: Header Option to always include the header with Log data notifications.	LDH	Obj\ENum Values: Auto, Always send
Destination Server Configure the destination URL and authentication	N	Fixed Container: On the Commander platform this will be *[CDM v20\JSONNotify v20\Network]* On the ObSys platform this will be *[OSM v20\JSONNotify v20\Network]*
Destination Available Indicates no failed attempts to send data	DS	Obj\NoYes
Last Notification Attempt Time of last attempt to contact server	LNT	Obj\DateTime
Last Response Code HTTP response code. A value in the range 200…206 indicates success	LRC	Obj\Num: 0, 100…505
Database Objects Available Count of maximum objects available from Essential Data and Extra Data	EDC	Obj\Num
Database Privilege Levels Configure privilege levels to control read access to Essential Data and Extra Data	P	Fixed Container: On the Commander platform this will be *[CDM v20\JSONNotify v20\Security]* On the ObSys platform this will be *[OSM v20\JSONNotify v20\Security]*
Debug Enable This will store additional debug information in the record file. Use this option only when instructed by North Support	DE	Obj\NoYes; Adjustable

Destination Server

Object Type: [OSM v20\JSONNotify v20\Network]
Object Type: [CDM v20\JSONNotify v20\Network]

The Destination Server contains the following objects.

Use the Server IP/DNS (IA), Server Port (PN) and Server Resource Name (RN) objects to specify the url of the endpoint server resource.

For example, if the url is http://test.northbt.com/api/webhooks/data?device=123456789&apiKey=ABCDE then set the objects as follows:

Server IP/DNS: test.northbt.com
Server Port: 80 (default HTTP port)
Server Resource Name: /api/webhooks/data?device=($sn)&apiKey=ABCDE

If you paste the whole url into object IA or RN, then it will be automatically separated into the relevant objects.

Description	Reference	Type
Server IP/DNS DNS or IP address of remote server	IA	Obj\Text; Max 64chars; Adjustable
Server Port TCP port number. Default 80.	PN	Obj\Num: 1…65535; Adjustable
Server Resource Name Name of endpoint file on server to send data. The following variables can be used: $(sn) – serial number of North device $(label) – System label $(type) – data type (json or csv) $(resync) – indicates sending all data after Essential Data structure change (1 or 0) $(date) – date (YYYYMMDD) $(time) – time (HHMMSS)	RN	Obj\Text; Max 120 chars; Adjustable
Server Timeout (secs) Maximum time to wait for a response from the remote server. Default 10 seconds.	TO	Obj\Num: 5…300; Adjustable
Authentication Type	ATY	Obj\ENum: 0…1; Adjustable 0=None, 1=HTTP Basic
Authentication Name Required when using HTTP Basic authentication	AID	Obj\Text; Max 40chars; Adjustable
Authentication Password Required when using HTTP Basic authentication	APW	Obj\Text; Max 40chars; Adjustable

Database Privilege Levels

Object Type: *[CDM v20\JSONNotify v20\Security]*
Object Type: *[OSM v20\JSONNotify v20\Security]*

Security Areas and Levels

Within the North security model, there are eight security areas. Security areas could be actual areas in a building, but are normally functional areas – for example, 'environmental control' and 'North engineering' areas would allow a user to have different privileges in controlling set points and engineering Commanders.

Typically, a user is assigned a privilege level in each of the eight areas. The level is in the range zero to seven, seven being the most powerful. When a user wishes to pass a door, his/her privilege level in the door's area is checked against the minimum required for that area – and then either allowed to pass, or rejected.

The engineer must decide the use of the eight areas. The engineer must also decide the power of the privilege levels. Most systems use only a few levels per area: 0=None, 1=Guest, 2=User, 7=Administrator.

As an example, imagine a page of values in Essential Data. The page needs a user to have a minimum privilege level of 2 in area 1 before it can be viewed. The page is available in a Web browser that checks users with a security database. User A has privilege level 7 in area 1 – she can view the page. User B has privilege level 5 in area 1 – he can also view the page. User C has privilege level 1 in area 1 – she cannot view the page.

The example continues: within this page of values in Essential Data is a temperature set point object. Users need a minimum privilege level of 6 in area 1 to adjust it – therefore User A can adjust the set point, but User B cannot.

Specifying Access Security

Essential Data and Extra Data have Access Security objects to control who can view a page, and who can adjust an adjustable object.

Each Access Security object has a two-digit value. Each controls the access to a particular feature - such as viewing the page, or adjusting the value. The two-digit value is made up of the area digit (1-8), followed by the minimum privilege level (1-7) – for example, if the minimum privilege level is 6 in area 2, then the two digit value is 26. If the value is 00, then no security checks are made.

JSONNotify Driver

The Database Privilege Levels object contains a privilege level for each of the eight security areas, representing a virtual user. The JSONNotify driver uses these to control access to Essential Data and Extra Data when reading a value.

Description	Reference	Type
Privilege Level in Area *x* The area, *x*, can be in the range 1…8	P*x*	Obj\Num; Adjustable; Range: 0…7

North JSON Notify

Object Type: [JSONNotify v20]

The North JSON Notify system contains a single object to trigger an alarm event. Refer to the notification type for the action performed.

Description	Reference	Type
Alarm Trigger Write any value to trigger the alarm message event	ALARM	Obj\Text; Adjustable

Driver Versions

Version	Build Date	Details
1.0	26/6/2014	Driver released
1.0	19/1/2015	Removed ':' separator from $(time) value as it is an invalid filename character.
2.0	1/9/2015	Updated to support Essential Data v3.0 and Extra Data Added objects Database Objects Available (EDC) and Database Privilege Levels (P.Px) Modified object references N.TY, N.SP, A.TY, A.ID, and A.PW (these are still supported)
2.0	24/11/2016	Fixed problem with Enum types in Essential Data Add driver object LDH.
2.0	01/11/2017	Optimized TCP/IP receive

Next Steps…

If you require help, contact support on 01273 694422 or visit *www.northbt.com/support*

North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

Author: JF
Checked by: BS

Document issued 08/11/2017.