



The JSONData Driver

The JSONData driver provides a web-based API to the North device's Essential Data and Extra Data, allowing you to collect and adjust information from it. Available for Commander and ObSys.

This document relates to JSONData driver version 2.0

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

Contents

Purpose of JSONData Driver	3
Values.....	3
Prerequisites.....	3
Detailed Operation	4
API Endpoint	4
Data Formats	4
API Methods	7
Object: Page List	7
Object: List	9
Object: Adjust.....	14
Device: List	16
Using the Driver	18
Starting the Interface	18
Setting up the Driver.....	18
Checking Communications	18
Alarms	19
Format.....	19
Examples.....	19
Condition Field	19
Object Specifications.....	20
Device Top-Level Objects	20
JSON Data Setup	21
Network.....	22
Network Interface.....	22
Security	23
Driver Versions	24

Purpose of JSONData Driver

The JSONData driver provides a web-based API to the North device's Essential Data and Extra Data, allowing you to request and adjust information from it.

Use this web service to integrate data collected using North interface technology directly into your own application. The service also supports cross-origin resource sharing (CORS), allowing your website to incorporate data too.

The JSONNotify driver is also available, sending data to an endpoint server using Webhooks.

Values

You can connect to the JSONData API using a simple HTTP request, and then receive a response in JSON format. The API 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.

The functions currently available from the API are:

- Object access functions – list values from the database and adjust them
- Device functions – retrieve device information

The JSONData driver can send alarms to the North system when a value is changed via the API.

Prerequisites

The Essential Data module should be configured, with access security levels set if required. The JSONData driver requires Essential Data v3.0 (build 01/09/2015) or later.

If authentication is required, add users to the Security Server module or use an API key.

Detailed Operation

API Endpoint

All JSONData API URLs listed in this documentation are relative to `http://address:port/api/`. For example, the `/device/list` method is reachable at `http://address:port/api/device/list`

Replace *address* with either the IP address or DNS name of the North device, and *port* with the API port number – for example `http://cmdr.northbt.com:81/api/device/list`

Configure the API port number using the *Network object* within the JSONData driver.

RESTful

The JSONData driver provides a RESTful API, with the following caveats:

- All API calls should be made with HTTP POST or HTTP GET
- All methods are accessed via `http://address:port/api/section/method`
- You can consider any non-200 HTTP response code an error

Security Considerations

To protect the API endpoint from unauthorised requests, there are two ways to identify your application:

- HTTP Basic authentication – authorize an individual request for data
- API key – a fixed key configured to allow access to a registered application

An additional option is also available that restricts the API to read-only access, prohibiting adjustments using the `object/adjust` method.

Configure all these options using the *Security object* within the JSONData driver.

The driver supports HTTP URLs only. HTTPS is not supported.

Data Formats

Request Format

A client sends an HTTP POST request with several parameters, as documented in each API method below. Parameters must be formatted as JSON objects with UTF-8 character encoding (application/json; charset=utf-8).

Here is a sample POST request showing header, and the body containing JSON parameters:

```
POST http://192.168.192.167:81/api/object/list HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 36

{
  "apiKey": "148F7A366C92B460",
  "oid": 32,
}
```

Alternatively, a client could send an HTTP GET with parameters in the URL:

```
GET http://192.168.192.167:81/api/object/list?apiKey=148F7A366C92B460&oid=32 HTTP/1.1
```

Response Format

The response format is JSON with UTF-8 character encoding (application/json; charset=utf-8). Each API method below describes the parameters included in the response.

JSONData will respond using HTTP 1.1 chunked transfer encoding.

Here is a sample response for the /device/list method:

```
HTTP/1.1 200 OK
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
  "status": "ok",
  "device": {
    "version": 1,
    "id": "80000000",
    "label": "Commander",
    "date": "2013-11-20T12:00:00",
    "revision": 1
  }
}
```

Error Response Format

Each response includes the status parameter, indicating an 'ok' or 'error'. For an error response, an error code and description are included along with a non-200 HTTP status code.

Body

```
{
  "status": "error",
  "eid": number,
  "message": string
}
```

Parameters

Parameter	Value	Description
status	String	Response status – 'error'
eid	Number	Error code – see list below
message	String	Text description of error

Error Codes

Error code (eid)	Message
0	HTTP error
1	JSON data too large
2	JSON data contains invalid character
3	JSON data incomplete
4	apiKey required
5	API method not recognised
6	Parameters missing
7	Object reference out-of-range
8	Object not adjustable

Example

```
HTTP/1.1 400 Bad request
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
```

```
{
  "status": "error",
  "eid": 5,
  "message": "API method not recognised"
}
```

API Methods

Object: Page List

This method gets the list of pages available from Essential Data and Extra Data.

Essential Data contains up to 64 pages on Commander, and 128 pages on ObSys. A page can be configured to contain 10, 16, 32, or 64 objects, with each object holding a single value. Extra Data contains 16 pages, with each containing 64 objects.

Provide an optional filter to retrieve a single page. If no filter is provided, then the response includes all pages available.

Request Format

URL

```
/object/page-list
```

Body

```
{
  "apiKey": string,
  "pid": number
}
```

Parameters

Parameter	Value	Description
apiKey	String	Optional API Key for your application
pid	Number	Optional filter to request only data for the specified page. Page identifier is in the range 1...128 for Essential Data (depending on platform), and 201...216 for Extra Data

Response Format

Body

```
{
  "status": "ok",
  "page": [
    {
      "pid": number,
      "label": string,
      "isUnreliable": boolean,
      "isOutOfRange": boolean,
    }, ...
  ]
}
```

Parameters

Parameter	Value	Description
status	String	Response status – ‘ok’ indicates success
page	Array	Array of pages from North device
pid	Number	Page identifier, in the range 1...128, or 201...216
label	String	Label of the page (max. 20 chars)
isUnreliable	Boolean	Summary of objects within page - ‘true’ indicates an object has a communications fault
isOutOfRange	Boolean	Summary of objects within page - ‘true’ indicates an object is out of range

Example

In this example, a pid is not specified in the request so the response includes all pages available from both Essential Data and Extra Data.

Request

```
POST http://192.168.192.167:81/api/object/page-list HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 0
```

Response

```
HTTP/1.1 200 OK
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
  "status": "ok",
  "page": [
    {
      "pid": 1,
      "label": "Server Environment",
      "isUnreliable": false,
      "isOutOfRange": false
    },
    {
      "pid": 2,
      "label": "UPS Status",
      "isUnreliable": true,
      "isOutOfRange": false
    },
    {
      "pid": 3,
      "label": "UPS Battery",
      "isUnreliable": false,
      "isOutOfRange": false
    },
    {
      "pid": 201,
      "label": "Fire Zone States",
      "isUnreliable": false,
      "isOutOfRange": false
    }
  ]
}
```


Object: List

This method gets a list of objects available from Essential Data and Extra Data.

Essential Data contains up to 640 objects on Commander, and 1280 objects on ObSys. Extra Data contains up to 1024 objects. Objects are organised into pages, with each page containing up to 64 objects.

Provide an optional filter to retrieve a page of objects, a single object, or reduce the amount of information returned. If no filter is provided, then the response includes all objects available.

Request Format

URL

```
/object/list
```

Body

```
{
  "apiKey": string,
  "oid": number,
  "pid": number,
  "lean": boolean
}
```

Parameters

Parameter	Value	Description
apiKey	String	Optional API Key for your application
oid	Number	Optional filter to request only data for specified object. Object identifier is in the range 1...1280 for Essential Data (depending on platform), and 2001...3024 for Extra Data
pid	Number	Optional filter to request only data from a specified page. Page identifier is in the range 1...128 for Essential Data (depending on platform), and 201...216 for Extra Data
lean	Boolean	Optional filter to respond with reduced information (true). Refer to response parameters below for those included

Response Format

Body

```
{
  "status": "ok",
  "obj": [
    {
      "oid": number,
      "type": string,
      "value": number | boolean | string |
        [{"s": string, "e": string}, ... ] |
        [{"s": string, "v": number}, ... ],
      "updated": string,
      "isUnreliable": boolean,
      "isOutOfRange": boolean,
      "label": string,
      "units": string,
      "hasAdjust": boolean,
      "maxValue": number,
      "minValue": number,
      "resolution": number,
      "enum": [{"i": number, "l": string}, ... ],
    }
  ]
}
```

```

    "maxPeriods": number,
    "pid": number,
    "pageLabel": string
  }, ...
]
}

```

Parameters

Parameter	Value	Description	Lean Response
status	String	Response status – ‘ok’ indicates success	Yes
obj	Array	Array of objects from North device	Yes
oid	Number	Object identifier, in the range 1...1280, or 2001...3024	Yes
type	String	Type of object, one of the following: text, noyes, offon, num, enum, float, datetime, date, times, or profile	Yes
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-ddThh: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)	Yes
s	String	Start time in 24hr format, ‘hh:mm’. Provided with Times and Profile types	Yes
e	String	End time in 24hr format, ‘hh:mm’. Only provided with times types	Yes
v	Number	Profile value. Only provided with profile types	Yes
updated	String	Date and time value last updated (local time) in ISO-8601 format, ‘yyyy-mm-ddThh:mm:ss’	Yes
isUnreliable	Boolean	If ‘true’, communications with the sub-system is failing	Yes
isOutOfRange	Boolean	If ‘true’, value is outside min/max limits	Yes
label	String	Label of the object (max. 20 chars)	No
units	String	Optional units for object value (max. 8 chars)	No
hasAdjust	Boolean	If ‘true’, value can be updated using /object/adjust method	No
maxValue	Number	Optional maximum value. Only provided with num and float types	No
minValue	Number	Optional minimum value. Only provided with num and float types	No
resolution	Number	Optional number of decimal places to display. Only provided with float types	No
enum	Array	Optional array of objects, each containing an index (i) and label (l) for each value. Only provided with enum types	No
i	Number	Index value of enumeration. Only provided with enum types.	No
l	String	Label for enumeration. Only provided with enum types.	No
maxPeriods	Number	Optional maximum number of on/off times or value/time profiles. Only provided with times or profile types	No
pid	Number	Page identifier, in the range 1...128, or 201...216	No
pageLabel	String	Label of the page (max. 20 chars)	No

Example

In this example, a pid is specified in the request so the response includes all objects available from the specified Essential Data page.

Request

```

POST http://192.168.192.167:81/api/object/list HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 19

```

```
{
  "pid": 2
}
```

Response

```
HTTP/1.1 200 OK
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
```

```
{
  "status": "ok",
  "obj": [
    {
      "oid": 17,
      "type": "float",
      "value": 12.3,
      "updated": "2013-12-31T23:40:16",
      "isUnreliable": false,
      "isOutOfRange": false,
      "label": "Load power",
      "units": "W",
      "hasAdjust": false,
      "resolution": 1,
      "pid": 2,
      "pageLabel": "UPS Status"
    },
    {
      "oid": 18,
      "type": "num",
      "value": 43,
      "updated": "2013-11-16T14:32:28",
      "isUnreliable": true,
      "isOutOfRange": false,
      "label": "Battery Time Left",
      "units": "mins",
      "hasAdjust": false,
      "pid": 2,
      "pageLabel": "UPS Status"
    },
    {
      "oid": 19,
      "type": "enum",
      "value": 1,
      "updated": "2013-12-31T23:40:16",
      "isUnreliable": false,
      "isOutOfRange": false,
      "label": "Battery",
      "hasAdjust": false,
      "enum": [{"i": 1, "l": "Healthy"}, {"i": 2, "l": "Replace"}],
      "pid": 2,
      "pageLabel": "UPS Status"
    },
    {
      "oid": 20,
      "type": "noyes",
      "value": false,

```

```

    "updated": "2013-12-31T23:40:16",
    "isUnreliable": false,
    "isOutOfRange": false,
    "label": "Test",
    "hasAdjust": true,
    "pid": 2,
    "pageLabel": "UPS Status"
  },
  {
    "oid": 21,
    "type": "text",
    "value": "Smart-UPS 450",
    "updated": "2013-12-31T23:32:27",
    "isUnreliable": false,
    "isOutOfRange": false,
    "label": "Model",
    "hasAdjust": false,
    "pid": 2,
    "pageLabel": "UPS Status"
  },
  {
    "oid": 22,
    "type": "datetime",
    "value": "2014-01-01T00:00:00",
    "updated": "2014-01-01T00:00:00",
    "isUnreliable": false,
    "isOutOfRange": false,
    "label": "Date",
    "hasAdjust": false,
    "pid": 2,
    "pageLabel": "UPS Status"
  },
  {
    "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,
    "isOutOfRange": false,
    "label": "Alarm",
    "hasAdjust": true,
    "maxPeriods": 2,
    "pid": 2,
    "pageLabel": "UPS Status"
  },
  {
    "oid": 28,
    "type": "profile",
    "value": [{"s": "00:00", "v": 18}, {"s": "09:00", "v": 21}],
    "updated": "2013-12-31T23:40:18",
    "isUnreliable": false,
    "isOutOfRange": false,
    "label": "Setpoint",
    "hasAdjust": true,
    "maxPeriods": 4,
    "pid": 2,
    "pageLabel": "UPS Status"
  }
}

```

```
]
}
```

Object: Adjust

Use this method to set the value of a single object in Essential Data or Extra Data. It is only available for objects with the `hasAdjust : true` property.

Control access to this method using authentication, or disable it from the *Security Setup object* within the JSONData driver.

Request Format

URL

```
/object/adjust
```

Body

```
{
  "apiKey": string,
  "oid": number,
  "value": number | boolean | string |
    [{"s": string, "e": string}, ... ] |
    [{"s": string, "v": number}, ... ]
}
```

Parameters

Parameter	Value	Description
apiKey	String	Optional API Key for your application
oid	Number	Object identifier returned from object/list. Object identifier is in the range 1...1280 for Essential Data (depending on platform), and 2001...3024 for Extra Data
value	Number, Boolean, String, or Array	New value of the 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-ddThh:mm:ss' Date – String in ISO-8601 format, 'yyyy-mm-dd' Times – Array of objects – each containing a start (s) and end (e) time. (If using an HTTP GET, the value parameter is in the format 'hh:mm-hh:mm, hh:mm-hh:mm') Profile – Array of objects - each containing a time (t) and value (v). (If using an HTTP GET. The value parameter is in the format 'hh:mm=value, hh:mm=value')
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 provide with profile types

Response Format

Body

```
{
  "status": "ok"
}
```

Parameters

Parameter	Value	Description
status	String	Response status – 'ok' indicates success

Example

In this example, the request sets Essential Data oid 17, of type float, with the new value '27.6'.

Request

```
POST http://192.168.192.167:81/api/object/adjust HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 39

{
  "oid": 17,
  "value": 27.6
}
```

Response

```
HTTP/1.1 200 OK
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
  "status": "ok",
}
```

Device: List

This method gets a list of summary information about the device.

Request Format

URL

```
/device/list
```

Body

```
{  
  "apiKey": string  
}
```

Parameters

Parameter	Value	Description
apiKey	String	Optional API Key for your application

Response Format

Body

```
{  
  "status": "ok",  
  "device": {  
    "version": number,  
    "id": string,  
    "label": string,  
    "date": string,  
    "revision": number  
  }  
}
```

Parameters

Parameter	Value	Description
status	String	Response status – ‘ok’ indicates success
device	Object	Device information
version	Number	API version – 1
id	String	Serial number of North device
label	String	North device label (max. 30 chars)
date	String	Current date and time (UTC) in ISO-8601 format
revision	Number	Essential Data/Extra Data revision number. This will increment when an object is added or amended within the database

Example

Request

```
POST http://192.168.192.167:81/api/device/list HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Content-Length: 0
```


Response

```
HTTP/1.1 200 OK
Server: JSONData/1.0.0 Commander/2.0
Date: Wed, 01 Jan 2014 00:00:00 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
```

```
{
  "status": "ok",
  "device": {
    "version": 1,
    "id": "99999999",
    "label": "Commander",
    "date": "2014-01-01T00:00:00Z",
    "revision": 2
  }
}
```

Using the Driver

On ObSys and Commander, the JSONData driver is pre-installed. Once started, you will need to set up the driver before you can make HTTP requests to it.

The JSONData driver uses zero licence units.

Starting the Interface

- 📖 To start an interface using the JSONData driver, follow these steps:
 - **Start Engineering** your North device using ObSys
 - Navigate to **Configuration, Interfaces**, and set an unused **Interface** to 'JSONData' 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 Data Setup**, should now be available.

Setting up the Driver

- 📖 To set up the driver, follow these steps:
 - Navigate to the **JSON Data Setup** object (Mc). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'
 - By default, the API is enabled at TCP port 81 on all available IP addresses. Navigate to the **Network** object (N) to change the port or force a single IP address
 - By default, authentication is disabled. Authentication can be enabled in the **Security** object (S) to use an API key or, once a user has been configured in the device's Security Server, HTTP authentication

Checking Communications

After configuring Essential Data, the driver will automatically provide information when requested. Check the **Server Open** object (DS) has a value 'yes'.

The **Values Available From** object (EDS) indicates if values are available to the API from Essential and Extra Data.

Before enabling security, make a request from a web browser to the API. For example, the `/device/list` method is reachable at `http://address:port/api/device/list`.

Replace *address* with either the IP address or DNS name of the North device, and *port* with the API port number. E.g. `http://192.168.192.167:81/api/device/list`

Alarms

When a value is set using the API object adjust method , the driver sends a North-format alarm to the device's alarm processing.

Format

North-format alarms contain six text fields. The JSONData driver places the following information into these fields:

System – copied from System Label object (DL) within driver setup

Point – contains, from Essential Data, *'page label – object label'*

Condition – see Condition Field section below

Priority – set using Value change alarm priority object (VC) within Security

Date & Time – from North device

Examples

System	Point	Condition	Priority	Date	Time
JSON Data	UPS - Test	1 by S Beckett	3	19/12/13	14:29:48
JSON Data	Heating - Setpoint	21 by E Bronte	3	20/12/13	14:26:26
JSON Data	Security - Enable	0 by A Turing	3	20/12/13	14:26:28
JSON Data	UPS Status - Alarm	Updated by A Lovelace	3	20/12/13	15:06:59
JSON Data	UPS Status - Alarm	Updated	3	28/12/13	11:42:31

Condition Field

The format of this field is: *'value user'*

value – contains the new value set. If the object adjusted is an array type, i.e. times or profile, then 'Updated' is used instead.

user – depends on the authentication method. If no authentication or API key methods are used, then this field is unused. Using HTTP authentication, this field contains the authenticated user's name in the format *'by user'*. E.g. *'by Ada Lovelace'*.

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 JSONData 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 Data Setup Set up the JSONData driver, started on interface <i>c</i> (<i>c</i> is the interface number)	Mc	Fixed Container: On the Commander platform this will be <i>[CDM v20\JSONData v20]</i> On the ObSys platform this will be <i>[OSM v20\JSONData v20]</i>

JSON Data Setup

Object Type: [OSM v20\JSONData v20]

Object Type: [CDM v20\JSONData v20]

The JSONData driver contains the following objects:

Description	Reference	Type
System Label Used when sending an alarm message	DL	Obj\Text: 20 chars; Adjustable
Enable Server	E	Obj\NoYes; Adjustable
Server Open Indicates that the API has successfully opened a port to listen for requests	DS	Obj\NoYes
Database Objects Available Count of maximum objects available from Essential Data and Extra Data	EDC	Obj\Num
Total Requests Total requests made via the API	RC	Obj\Num
Network Configure the port number and IP address the API is available at	N	Fixed Container: On the Commander platform this will be <i>[CDM v20\JSONData v20\Network]</i> On the ObSys platform this will be <i>[OSM v20\JSONData v20\Network]</i>
Security Control access to the API	S	Fixed Container: On the Commander platform this will be <i>[CDM v20\JSONData v20\Security]</i> On the ObSys platform this will be <i>[OSM v20\JSONData v20\Security]</i>
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

Network

Object Type: [OSM v20\JSONData v20\Network]

Object Type: [CDM v20\JSONData v20\Network]

Configure the JSONData Network connectivity using this object. By default, all available IP addresses are opened for requests to the API on TCP port 81.

If required, change the TCP port number or restrict access to a single IP address.

Description	Reference	Type
Interfaces open Reports the number of interfaces the API is available on	C	Obj\Num: 0...8
Force single IP address Force the driver to use only one of the interface IP addresses. By default, all are used when the address is '0.0.0.0'	IA	Obj\IP; Adjustable
TCP Port TCP port number. Default 81	PN	Obj\Num: 1...65535; Adjustable
Interface x Status information for a network interface available on the North device. The interface number, x, is in the range 1...4 on Commander and 1...8 on ObSys	Ix	Fixed Container: On the Commander platform this will be [CDM v20\JSONData v20\Network\Interface] On the ObSys platform this will be [OSM v20\JSONData v20\Network\Interface]

Network Interface

Object Type: [OSM v20\JSONData v20\Network\Interface]

Object Type: [CDM v20\JSONData v20\Network\Interface]

An Interface represents a physical or virtual network interface on the platform. Use this object to find out the IP address available and if JSONData has opened a port to listen for requests.

Description	Reference	Type
IP address IP address available for the interface	IA	Obj\IP
Port open Indicates if the API has opened a port on the interface	S	Obj\NoYes

Security

Object Type: [OSM v20\JSONData v20\Security]

Object Type: [CDM v20\JSONData v20\Security]

Enable security for the API using this object. Objects are available to require authentication from an application, and enable read-only access.

The Default Database Privilege Levels object contains a privilege level for each of the eight security areas. The JSONData driver uses this virtual user to control access to Essential Data and Extra Data when HTTP authentication is not used.

Description	Reference	Type
Authentication Method Select authentication method required for API requests	M	Obj\Enum; Adjustable Values: None, HTTP, API Key
Security Server Object Object reference of North Security Server. Only required authentication method is set to HTTP	RO	Obj\Obj; Adjustable
API Key When authentication method is set to API key, specify the value of the key using this object	AK	Obj\Text: Length 16-32 chars; Adjustable Set a blank value to generate a new random key
Enable Value change Enable write access to API using the /object/adjust method	WE	Obj\NoYes; Adjustable
Value change alarm priority When a value is adjusted from the API, generate an alarm	VC	Obj\Num: 0...9; Adjustable Use 0 to disable change alarms
Default Privilege Level in Area x The area, x, can be in the range 1...8. Not used when authentication method is set to HTTP	Px	Obj\Num; Adjustable; Range: 0...7

Driver Versions

Version	Build Date	Details
1.0	10/2/2014	Driver released
1.0	25/8/2015	Resolved issue in writing to a Profile object with more than three time/value pairs
2.0	02/9/2015	Updated to support Essential Data v3.0 and Extra Data driver. Added default privilege levels for accessing Essential Data/Extra Data.

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

This document is subject to change without notice and does not represent any commitment by North Building Technologies Ltd.

ObSys and Commander are trademarks of North Building Technologies Ltd. All other trademarks are property of their respective owners.

© Copyright 2016 North Building Technologies Limited.

Author: JF
Checked by: BS

Document issued 11/03/2016.