

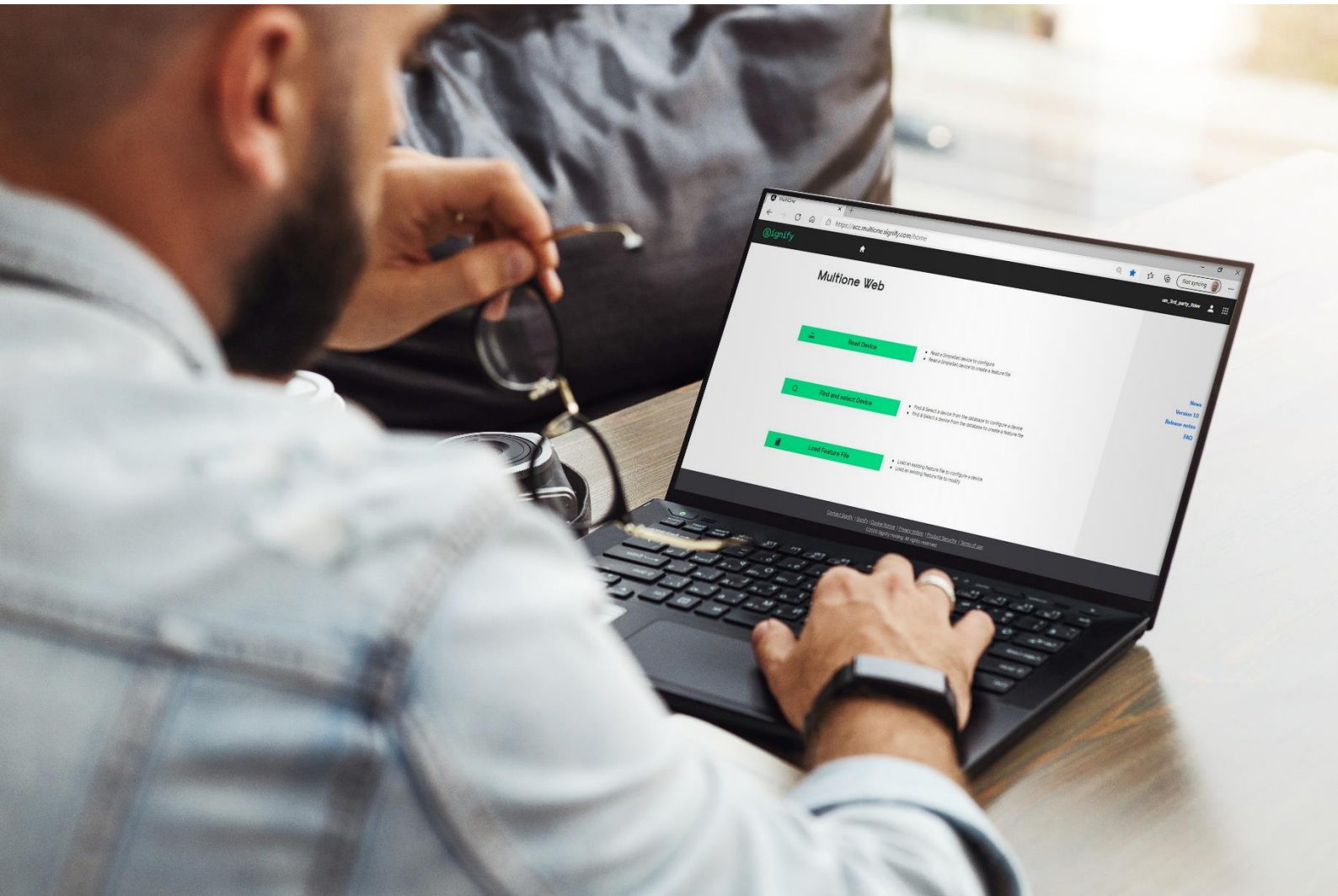


# MultiOne

by  **signify**

MultiOne Cloud  
Configuration tooling

## MultiOne Cloud 1.14.1



### Application Guide for MultiOne Cloud Document version March 2024

## About this document

This document describes

1. How to connect to MultiOne Cloud using Postman
2. The API (Application Programming Interface) calls for the generation of feature files
3. Example syntax for all driver features useful for configuration

## Scope and Purpose

MultiOne cloud features and use of REST APIs (REpresentational State Transfer Application Programming Interface) with Postman application that supports your automation system.

## Intended Audience

This document is intended for software experts and developers of the OEM who are involved to the fullest in the integration of MultiOne Cloud in the Automation systems.

For access, contact [multione@signify.com](mailto:multione@signify.com) (or) regional Key Account Managers.

## Privacy notes and Terms Agreement

For the privacy notes for MultiOne Cloud and MultiOne Web visit our site:

<https://www.signify.com/global/privacy/legal-information/privacy-notice>

The term of conditions for these tools can be found on:

<https://www.signify.com/global/terms-of-use>

# Version History

Date	Changes
22 Oct 2020	Added items SR Power Removed in feature Dimming Interface “AmpDim”, since this is not yet supported Minor textual changes
22 Nov 2021	Added comment on Override possibilities for the Dimming Interface Minor textual change
15 Mar 2022	Encryption of feature file removed API commands modified to support multiple calls in one session Added features: <ul style="list-style-type: none"><li>- 0-10V interface</li></ul> Removed features: <ul style="list-style-type: none"><li>- Corridor Mode</li><li>- Touch and Dim</li></ul>
11 Jun 2022	Added features Beta <ul style="list-style-type: none"><li>- MidnightShift calculator</li><li>- MTP (Module Temperature Protection)</li><li>- AmpDim</li></ul> Note – In this version ‘MidpointShift’ is changed to ‘MidnightShift’ The version is backward compatible  Works for Production url and Beta url (See supported device List for June 2022) <a href="https://api.multione.signify.com/auth/api/v1">https://api.multione.signify.com/auth/api/v1</a> <a href="https://api.beta.multione.signify.com/auth/api/v1">https://api.beta.multione.signify.com/auth/api/v1</a>
13 Dec 2022	API call for list of device features Added features <ul style="list-style-type: none"><li>- Corridor Mode</li><li>- Touch and Dim</li><li>- OWP (OEM Write Protection)</li><li>- Luminaire info Format 0</li></ul> The version is backward compatible Works only for Production url (See supported device List for December 2022) <a href="https://api.multione.signify.com/auth/api/v1">https://api.multione.signify.com/auth/api/v1</a> No changes were made in the Beta environment  Note (1) All the feature names are case insensitive. For E.g., Feature name ‘AOC’ can be written as aoc, Aoc Feature name ‘DimmingInterface’ can be written as dimminginterface, dimmingInterface Note (2) Under Dimming Interface, feature parameters of Selected Interface and Override Interface are case sensitive – Refer the Dimming Interface Section Note (3) Under feature name OWP/Owp/owp, feature parameters are case sensitive Refer the OWP section

16 Aug 2023

Updated features

- AOC (Adjustable Output Current)
- Midnight Shift Calculator
- Driver Temperature Limit
- Luminaire Info format 3 (DALI part 251)
- AST (Adjustable Start-up Time)
- ZTV (Zero to Ten Volt)
- Corridor
- Touch and Dim
- OWP (OEM Write Protection)
- Dynadimmer

*The feature name "Dynadim" is deprecated. It is replaced with "Dynadimmer"*

Added features

- Retrieve driver production info
- Coded Mains

1 Jan 2024

Updated features

- MTP

Added features

- DALI 253M

Added Features Metadata API call

3 Mar 2024

Updated features

- SR Power/DALI Power supply  
*The feature name "SRPSU" is deprecated. It is replaced with "DALIPSU"*
- Luminaire info Format 3

New search API call to present EDIT information

## Table of Contents

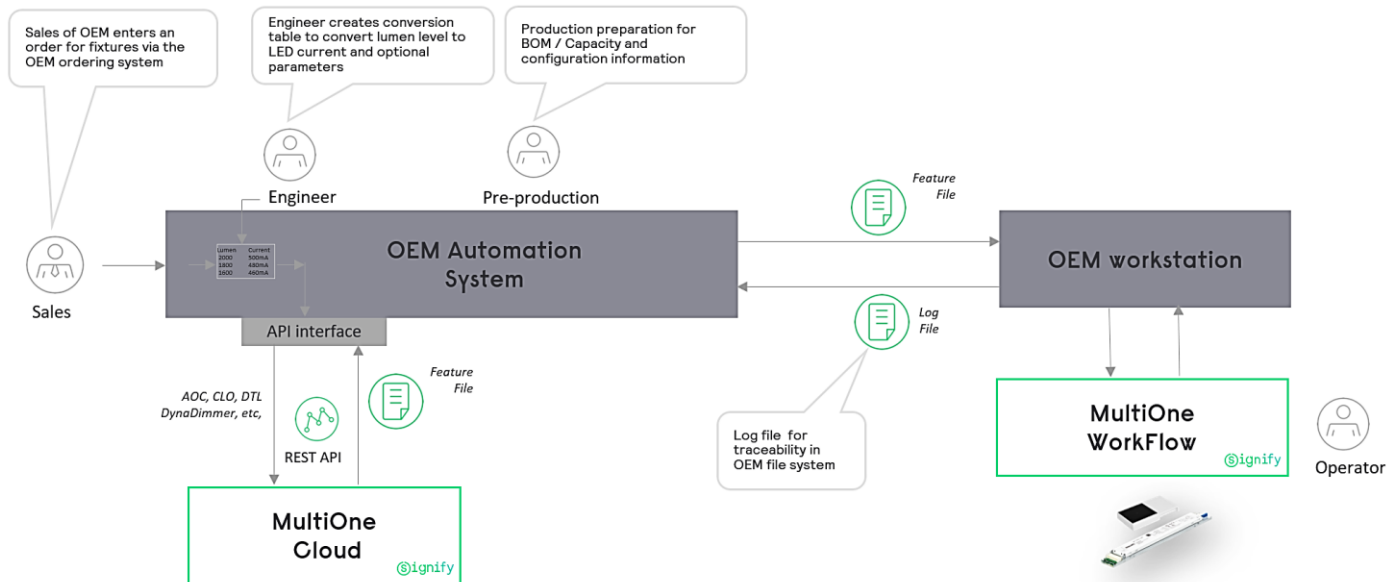
About this document.....	2
Version History.....	3
Introduction.....	7
API sequence for MultiOne.....	8
API calls for MultiOne Cloud.....	9
Authentication .....	10
Configuration .....	10
Supported features .....	11
Feature configuration.....	12
AOC – Adjustable Output Current.....	12
ALO (Adjustable Light Output).....	13
CLO (Constant Light Output).....	14
CLOlite (Constant Light Output Lite) .....	15
Coded Mains.....	16
Dimming Interface .....	18
Dynadimmer.....	19
Midnight Shift Calculator.....	21
DC Emergency.....	22
Minimum Dim Level.....	22
LineSwitch .....	23
DALI Power Supply Unit (DALIPSU).....	24
Driver Temperature Limit.....	25
Luminaire Info format 3 (DALI part 251).....	26
AST (Adjustable Start-up Time).....	28
ZTV (Zero to Ten Volt).....	28
AmpDim.....	29
MTP (Module Temperature Protection).....	30
Corridor .....	32
Touch and Dim.....	33
DALI 253M.....	34
OWP (OEM Write Protection).....	35
Multiple feature configuration with a single API call .....	36
Feature file Generation .....	37
References.....	38

# List of Tables

- Table 1 : Main API Calls.....9
- Table 2 : Additional API Calls.....9
- Table 3 : Feature call for AOC – Adjustable Output Current..... 12
- Table 4: Feature call for ALO (Adjustable Light Output) ..... 13
- Table 5: Feature call for CLO – Constant Light Output..... 14
- Table 6: Feature call for CLOlite – Constant Light Output Lite ..... 15
- Table 7: Feature call for Coded Mains ..... 16
- Table 8: Feature call for Dimming Interface..... 18
- Table 9: Feature call for Dynadimmer ..... 19
- Table 10: Feature call for Midnight Shift Calculator..... 21
- Table 11: Feature call for DC Emergency ..... 22
- Table 12: Feature call for Minimum Dim Level..... 22
- Table 13: Feature call for Lineswitch ..... 23
- Table 14: Feature call for DALI Power Supply Unit ..... 24
- Table 15: Feature call for Driver Temperature Limit..... 25
- Table 16: Feature call for Luminaire Info format 3 (DALI part 251)..... 26
- Table 17: Feature call for AST- Adjustable Start-up Time..... 28
- Table 18: Feature call for ZTV- Zero to Ten Volt..... 28
- Table 19: Feature call for AmpDim ..... 29
- Table 20: Feature call for MTP – Module Temperature Protection..... 30
- Table 21: Feature call for Corridor ..... 32
- Table 22: Feature call for Touch and Dim..... 33
- Table 233: Feature call for DALI 253M ..... 34
- Table 244: Feature call for OWP – OEM Write Protection..... 35

# Introduction

MultiOne Cloud can decrease time to market and optimize flexibility by connecting your automated planning/resource system. This new tool generates feature files compatible with MultiOne Engineering and MultiOne Workflow that is used in production. It also supports the latest configurable drivers sold from 2022 onwards. MultiOne Cloud is built on the open standard of REST API standardization in JavaScript Object Notation (JSON) format which makes it accessible via many different platforms. MultiOne Cloud is generally used by OEMs who have been using an automation system or planning to implement it. There are also other systems for e.g., excel etc. to which REST API can be connected.



The figure above represents the process flow. The sales personnel/technician enters the sales order, and the details are then converted to driver specific information. Based on this information in the automation system, the system makes specific API calls to connect with MultiOne Cloud as described in this document. In this way the feature files can be generated automatically and can be used in the production for configuring drivers.

APIs, short for Application Programming Interfaces, are software-to-software interfaces. Meaning, they allow different applications to talk to each other and exchange information or functionality. This allows businesses to access another business's data, piece of code, software, or services to extend the functionality of their own products – all while saving time and money (HubSpot, 2022).

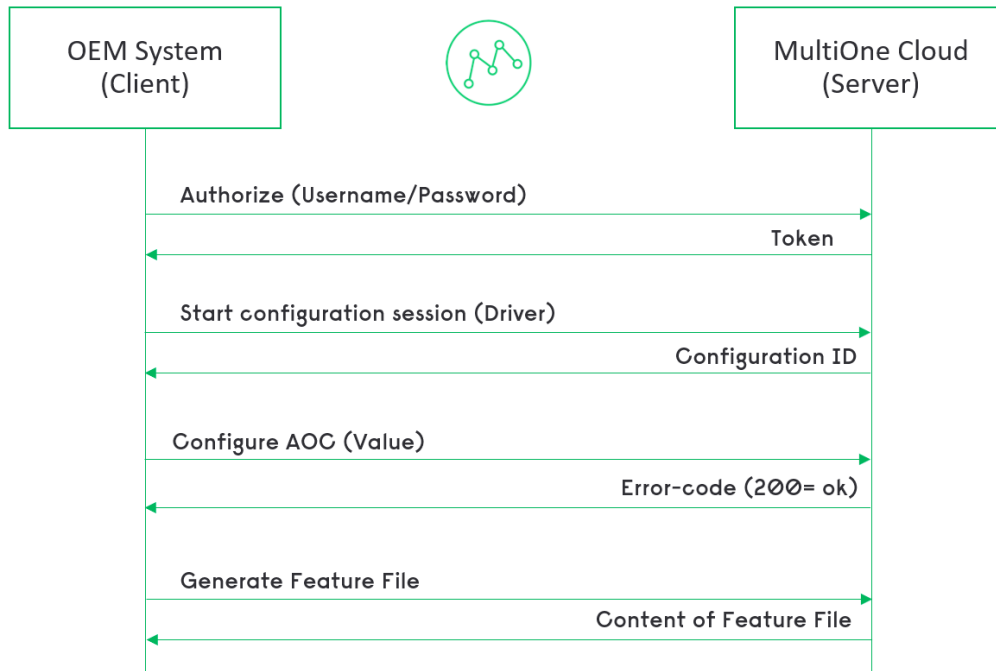
An API call is the process of a client application submitting a request to an API and that API retrieving the requested data from the external server or program and delivering it back to the client.

Request for the API calls for MultiOne Cloud are done in the following ways:

- **POST:** To authorize, to initiate configuration and configuration of features
- **GET:** To get the software version, list of device features and content of the feature file from MultiOne Cloud

There are many API testing platforms and one of the most common applications is Postman. This tool can be used for making API calls and in this document, the examples are provided using this tool.

## API sequence for MultiOne



As shown in the above figure, the following steps are needed to create a feature file:

- Authentication (of the registered user) – MultiOne Cloud checks if the user is authorized to communicate with the server. When that is passed, a token is returned.
- Start a configuration session – For every driver configuration, a configuration ID is returned. Multiple sessions are allowed with same authentication.
- Set the feature configuration(s) – Configure features for the driver. Depending on the set values, error codes are returned
- Generate the feature file – After the configuration of features, via ‘Get’ feature file, you can get the content of the feature file. This content can be stored in your system for further use. Note – After the content has been stored, you cannot make changes to the content. You need to start a new session.

All the supporting documents for MultiOne Cloud can be found on Global OEM tools portal/My Technology Portal.

Links : [Login | Tools \(signify.com\)](#) (or) [Login | My Technology Portal EMEA \(signify.com\)](#)



# API calls for MultiOne Cloud

Functions		API call
Authentication	The user account is authenticated, and a token is returned	<a href="#">{{url}}/auth/api/v1/authenticate</a>
Start configuration	A configuration ID is generated for the entered 12NC	<a href="#">{{url}}/config/api/v1/configuration</a>
Configure features	The features can be configured	<a href="#">{{url}}/config/api/v1/configuration/{{configurationId}}/features</a>
Feature File	Feature file generated and stored by OEM.	<a href="#">{{url}}/feature/api/v1/configuration/{{configurationId}}/feature-file</a>

Table 1 : Main API Calls

Functions		API Calls
Device Features	To get the list of device feature of the 12NC provided	<a href="#">{{url}}/search/api/v1/device/{{device12nc}}/features</a>
Device Features Metadata	To get the details of device features of the 12NC provided	<a href="#">{{url}}/search/api/v1/device/{{device12nc}}/features/metadata</a>
Midnight shift Calculator	To get the midnight shift value	<a href="#">{{url}}/config/api/v1/calculation/api/v1/midnight-shift</a>
Retrieve driver production info	To get the 12nc, device name and production related information.	<a href="#">{{url}}/search/api/v1/data/devices/{{deviceId}}/production-information</a>
Search API call	Present EDIT information	<a href="#">{{url}}/search/api/v1/devices/</a>

Table 2 : Additional API Calls

To select beta or production environment it is advised to define the url for MultiOne Cloud as a variable:

Production:

[{{url}}](#) – <https://api.multione.signify.com/>

Tip – When you make an API call to retrieve the list of device features, you can use the syntax from the response for easy implementation.

## Authentication

With your account credentials you can start the authentication process.

```
{  
  "password": "{{secret}}",  
  "userName": "{{userId}}"  
}
```

EXAMPLE

```
{  
  "password": "MultiOneCloud",  
  "userName": "multione"  
}
```

## Configuration

```
{  
  "device12nc": "String",  
}
```

EXAMPLE

```
{  
  "device12nc": "929002102906",  
}
```

## Supported features

- AOC (Adjustable Output Current)
- CLO (Constant Light Output)
- CLO Lite
- Dimming Interface selection
- ALO (Adjustable Light Output)
- ALO min
- Minimum dimlevel (different from 0-10V/1-10V dimlevel)
- DC emergency
- Luminaire info (DALI: part 251)
- DynaDimmer
- DTL (Driver temperature limit)
- LineSwitch
- SR/DALI power supply (DALI: part 250)
- 0-10V/1-10V Interface
- AST (Adjustable Startup Time)
- Midnight shift calculator
- MTP (Module Temperature Protection)
- AmpDim
- Corridor Mode
- Touch and Dim
- Coded Mains
- DALI 253M
- OWP (OEM Write Protection)

# Feature configuration

Below you can find some examples for configuration of features. See the design in guides of the driver for the available features. The syntax of the features is slightly changed (in accordance with REST definition). Please find below examples of the code. All the examples syntax is taken from Postman.

The features have in general the following syntax.

```
{
  "featureName": "String",
  "featureParameters": [
    {
      "Featureparameter1": Boolean,
      " Featureparameter2": Integer,
      " Featureparameter3": "String"
    }
  ]
}
```

## *AOC – Adjustable Output Current*

With the Adjustable Output Current (AOC) feature you can adjust the output current to a desired optimum value.

featureName	featureParameter	Type	Remark
AOC		String	Case Insensitive
	AocEnabled	Boolean	true/false
	AocCurrent	Integer	Value in mA Driver Specific

Table 3 : Feature call for AOC – Adjustable Output Current

### EXAMPLE

```
[
  {
    "featureName": "AOC",
    "featureParameters": [
      {
        "AocEnabled": true,
        "AocCurrent": 300
      }
    ]
  }
]
```

## ALO (Adjustable Light Output)

Use the Adjustable Light Output (ALO) feature to regulate the light output to a desired optimum value. You can create a “virtual lamp” with output different than what the standard wattages offer.

featureName	featureParameter	Type	Remark
ALO		String	Case Insensitive
	AloEnabled	Boolean	true/false
	AloPercentage	Integer	Value in %
	AloMinLevel	Integer	Value in % Driver Specific

Table 4: Feature call for ALO (Adjustable Light Output)

### EXAMPLE

```
[
  {
    "featureName": "ALO",
    "featureParameters": [
      {
        "AloEnabled": true,
        "AloPercentage": 80,
        "AloMinLevel": 10
      }
    ]
  }
]
```

## CLO (Constant Light Output)

The lumen output of a LED module usually decreases over its lifetime. To compensate, a schedule can be created that gradually increases the lamp's power level over time.

featureName	FeatureParameter	Type	Remark
CLO		String	Case Insensitive
	CloEnabled	Boolean	true/false
	CloPointx	X = Integer	Min 1 and Max 16 points
	Time	Integer	0 – 120 value in *1000 hours
	Percentage	Integer	% Of current

Table 5: Feature call for CLO – Constant Light Output

### EXAMPLE

```
[
  {
    "featureName": "CLO",
    "featureParameters": [
      {
        "CloEnabled": true,
        "CloPoint1": {
          "Time": 0,
          "Percentage": 80
        },
        "CloPoint2": {
          "Time": 50,
          "Percentage": 90
        },
        "CloPoint3": {
          "Time": 100,
          "Percentage": 100
        }
      }
    ]
  }
]
```

## CLOlite (Constant Light Output Lite)

The lumen output of a LED module usually decreases over its lifetime. To compensate, a schedule can be created that gradually increases the lamp's power level over time.

featureName	FeatureParameter	Type	Remark
CLOlite		String	Case Insensitive
	CloLiteEnabled	Boolean	true/false
	CloLiteInitialPowerLevel	Integer	Value in %
	CloLiteMaximumWorkingHours	Integer	0 – 120 value in *1000 hours

Table 6: Feature call for CLOlite – Constant Light Output Lite

### EXAMPLE

```
[
  {
    "featureName": "CLOLITE",
    "featureParameters": [
      {
        "CloLiteEnabled": true,
        "CloLiteInitialPowerLevel": 90,
        "CloLiteMaximumWorkingHours": 100
      }
    ]
  }
]
```

## Coded Mains

Coded Mains is the feature to configure the coded mains settings that are used when a coded mains command is received from a transmitter regarding the requested dimming scene for the driver.

featureName	FeatureParameter	Type	Remark
CodedMains		String	Case Insensitive
	FadeTime	Integer	0 - 15
	FastFadeTime	Integer	0 - 27
	FallbackDetectionTime	Integer	1 - 255
	PowerOnLevel	Integer	0 - 100 %
	FallbackScene	Integer	0 - 100 %
	Scenex	Integer	x = 1 - 10 max 10 scenes are allowed 0 - 100 %

Table 7: Feature call for Coded Mains

### EXAMPLE

```
[
  {
    "featureName": "CodedMains",
    "featureParameters": [
      {
        "FadeTime": 7,
        "FastFadeTime": 27,
        "FallbackDetectionTime": 5,
        "PowerOnLevel": 100,
        "FallbackScene": 100,
        "Scene1": 0,
        "Scene2": 1,
        "Scene3": 10,
        "Scene4": 35,
        "Scene5": 50,
        "Scene6": 60,
        "Scene7": 75,
```



```
"Scene8": 80,  
"Scene9": 85,  
"Scene10": 90,  
"Scene11": 95,  
"Scene12": 100  
}  
]  
}  
]
```

## Dimming Interface

Use the Dimming Interface feature to select the dimming interface for the device. The actual options are depending on the device model

featureName	FeatureParameter	Type	Remark
DimmingInterface		String	Case Insensitive
	SelectedInterface	String	Case Sensitive <a href="#">Dynadimmer</a> , <a href="#">AmpDim</a> , <a href="#">LineSwitch</a> , <a href="#">ZTV</a> and <a href="#">Dali</a>
	OverrideInterface	String	Only Dynadimmer as SelectedInterface can be overridden by: "LineSwitch"  Case Sensitive String <a href="#">LineSwitch</a> , <a href="#">ZTV</a>  OverrideInterface can contain multiple interfaces

Table 8: Feature call for Dimming Interface

### EXAMPLE

```
[
  {
    "featureName": "DimmingInterface",
    "featureParameters": [
      {
        "SelectedInterface": "Dynadimmer",
        "OverrideInterface": ["LineSwitch"]
      }
    ]
  }
]
```

## Dynadimmer

Dynadimmer is an autonomous dimmer for outdoor applications. You can create a dynamic dimming schedule over a period.

featureName	FeatureParameter	Type	Remark
Dynadimmer		String	Case Insensitive
	DynadimmerEnabled	Boolean	Optional - only when driver does not have interface selection
	MidnightShift	Integer	the midnight shift (value in minutes (0)) depends on the geographical location which can be calculated by a separate API call (Refer Midnight Shift Calculator section below)
	TimeReference	Boolean	true/false
	Scenex	Integer	x = 1 - 5 max 5 scenes are allowed
	Hour	Integer	1 - 24
	Minute	Integer	1 - 60
	DimLevel	Integer	1 - 100 % Value in %
	FadeTime	Integer	0 - 255 seconds
	OverrideMode	Boolean	Optional; only when driver does not have interface selection

Table 9: Feature call for Dynadimmer

### EXAMPLE

```
[
  {
    "featureName": "Dynadimmer",
    "featureParameters": [
      {
        "DynadimmerEnabled": false,
        "MidnightShift": 0,
        "TimeReference": false,
        "Scene1": {
          "DimLevel": 50,
          "FadeTime": 180,
          "Minute": 0,
```

```
    "Hour": 0
  },
  "Scene2": {
    "DimLevel": 100,
    "FadeTime": 180,
    "Minute": 0,
    "Hour": 6
  }
]
}
```

## Midnight Shift Calculator

The midnight shift depends on the geographical location. Enter the feature parameters to calculate the midnight shift in minutes. You can use the response value as input to configure Dynadimmer (Refer Dynadimmer section)

featureName	featureParameter	Type	Remark
	latitude	Integer	Degrees cannot be negative
	degrees	Integer	0 – 89
	minutes	Integer	0-60
	seconds	Integer	0-60
	hemisphere	String	North/South/East and West
	longitude	Integer	Degrees cannot be negative
	degrees	Integer	0 – 179
	minutes	Integer	0-60
	seconds	Integer	0-60
	hemisphere	String	North/South/East and West
	timeZone	Integer	GMT+1 enter 1, for GMT-1 enter -1, and so on.
	daylightSaving	Boolean	true/false

Table 10: Feature call for Midnight Shift Calculator

### EXAMPLE

```
{
  "latitude": {
    "degrees": 1,
    "minutes": 0,
    "seconds": 0,
    "hemisphere": "North"
  },
  "longitude": {
    "degrees": 1,
    "minutes": 0,
    "seconds": 0,
    "hemisphere": "West"
  },
  "timeZone": 0,
  "daylightSaving": false
}
```

## DC Emergency

This feature defines the behaviour of the driver when DC (emergency) is applied.

featureName	FeatureParameter	Type	Remark
DcEmergency		String	Case Insensitive
	DcEmergencyLevel	Integer	157 - 254 85 ~1%, 25 ~100%
	DcEmergencyEnabled	Boolean	true/false
	DcEmergencyDimmingEnabled	Boolean	(MO: "Allow dimming")
	LevellnArcPower	Boolean	true/false

Table 11: Feature call for DC Emergency

### EXAMPLE

```
[
  {
    "featureName": "DcEmergency",
    "featureParameters": [
      {
        "DcEmergencyLevel": 157,
        "LevellnArcPower": false,
        "DcEmergencyEnabled": true,
        "DcEmergencyDimmingEnabled": true
      }
    ]
  }
]
```

### Minimum Dim Level

This feature defines the minimum dimlevel.

featureName	FeatureParameter	Type	Remark
MinDimLevel		String	Case Insensitive
	Enabled	Boolean	true/false
	MinDimLevel	Integer	Value in %

Table 12: Feature call for Minimum Dim Level

### EXAMPLE

```
[
  {
    "featureName": "MinDimLevel",
    "featureParameters": [
      {
        "Enabled": true,
        "MinDimLevel": 10
      }
    ]
  }
]
```

## LineSwitch

This feature defines the behaviour when an external voltage is supplied to the line switch connector.

featureName	FeatureParameter	Type	Remark
LineSwitch		String	Case Insensitive
	HighVLevel	Integer	0 - 100 Percentage
	LowVlevel	Integer	0 - 100 Percentage
	FadeUpTime	Integer	0 - 90 Seconds
	SwitchOffDelayTime	Integer	0 - 1800 Seconds
	FadeDownTime		
	LineswitchEnabled	Boolean	Optional. When for DimmingInterface: Lineswitch is selected: LineswitchEnabled = true
	2ndInativeDimLevel	Integer	0 - 100 Percentage
	2ndInactiveLevelDelayTime	Integer	0 - 1440 minutes
	2ndInativeLevelEnabled	Boolean	true/false

Table 13: Feature call for Lineswitch

### EXAMPLE

```
[
  {
    "featureName": "LineSwitch",
    "featureParameters": [
      {
        "HighVLevel": 100,
        "LowVLevel": 50,
        "FadeUpTime": 0,
        "SwitchOffDelayTime": 0,
        "FadeDownTime": 0,
        "2ndInactiveDimLevel": 0,
        "2ndInactiveLevelDelayTime": 0,
        "2ndInactiveLevelEnabled": true
      }
    ]
  }
]
```

## DALI Power Supply Unit (DALIPSU)

With this feature the power for an external device can be switched on or off.

featureName	FeatureParameter	Type	Remark
DALIPSU		String	Case Insensitive
	Enabled	Boolean	true/false
	HbEnable	Boolean	Optional, driver specific true/false
	LastGaspEnable	Boolean	Optional, driver specific true/false

Table 14: Feature call for DALI Power Supply Unit

### EXAMPLE

```
[
  {
    "featureName": "DALIPSU",
    "featureParameters": [
      {
        "Enabled": true,
        "HbEnable": true,
        "LastGaspEnable": false
      }
    ]
  }
]
```



## Driver Temperature Limit

This feature defines the behaviour of the driver in case of high internal temperatures.

featureName	FeatureParameter	Type	Remark
Dtl		String	Case Insensitive
	DtlOutputPercentage	Integer	0-100
	DtlDimStart	Integer	Driver Specific
	DtlDimStop	Integer	Optional Driver Specific
	DtlLampOff	Integer	Optional Driver Specific
	DtlEnabled	Boolean	Optional true/false

Table 15: Feature call for Driver Temperature Limit

### EXAMPLE

```
[
  {
    "featureName": "Dtl",
    "featureParameters": [
      {
        "DtlEnabled": true,
        "DtlDimStart": 70,
        "DtlDimStop": 90,
        "DtlLampOff": 98,
        "DtlOutputPercentage": 60
      }
    ]
  }
]
```

## Luminaire Info format 3 (DALI part 251)

With this feature information can be set of the luminaire.

featureName	FeatureParameter	Type	Remark
LumInfoFormat		String	Case Insensitive
	OemGtin	String	13 digits
	OemIdentification	String	
	UseDeviceUid	Boolean	true/false
	ContentIdentifier	Integer	0 and 3
	LuminaireManufactureYear	Integer	1 - 99
	LuminaireManufactureWeek	Integer	1 - 99
	LuminaireManufactureYear	Integer	1 - 99
	Use DynamicDateIndicator	Boolean	true/false
	NominallInputPower	Integer	0 - 65535
	PowerMinDim	Integer	0 - 65535
	MinAcMainsVoltage	Integer	0 - 65535
	MaxAcMainsVoltage	Integer	0 - 65535
	NominalLightOutput	Integer	0 - 16777215
	CRI	Integer	0 - 255
	CCT	Integer	0 - 65535
	DistributionType	Integer	0 - 255
	LuminaireColor	String	This is a String of hexadecimal ascii values. Only the printable ascii character are allowed.
	LuminaireIdentification	String	This is a String of hexadecimal ascii values. Only the printable ascii character are allowed.
	AdditionalInfo	String	This is a String of hexadecimal ascii values. Only the printable ascii character are allowed.

Table 16: Feature call for Luminaire Info format 3 (DALI part 251)

## Example

```
[
  {
    "featureName": "LumInfoFormat",
    "featureParameters": [
      {
        "ContentIdentifier": 3,
        "OemGtin": "99999999999999",
        "OemIdentification": "",
        "UseDeviceUid": true,
        "LuminaireManufactureYear": 22,
        "LuminaireManufactureWeek": 40,
        "UseDynamicDateIndicator": true,
        "NominalInputPower": 0,
        "PowerMinDim": 0,
        "MinAcMainsVoltage": 194,
        "MaxAcMainsVoltage": 264,
        "NominalLightOutput": 3000,
        "CRI": 90,
        "CCT": 2700,
        "DistributionType": 2,
        "LuminaireColor": "",
        "LuminaireIdentification": "",
        "AdditionalInfo": ""
      }
    ]
  }
]
```

## AST (Adjustable Start-up Time)

This feature defines the behaviour of the driver in case of high internal temperatures.

featureName	FeatureParameter	Type	Remark
AST		String	Case Insensitive
	AdjustableStartupTime	Integer	0 - 1000

Table 17: Feature call for AST- Adjustable Start-up Time

### EXAMPLE

```
[
  {
    "featureName": "AST",
    "featureParameters": [
      {
        "AdjustableStartupTime": 700
      }
    ]
  }
]
```

## ZTV (Zero to Ten Volt)

This feature defines the behaviour of the driver in case of high internal temperatures.

featureName	FeatureParameter	Type	Remark
ZTV		String	Case Insensitive
	ZtvMinDimLevel	Integer	5 - 100
	ZtvCurveSelection	Integer	Driver specific
	ZtvEnabled	Boolean	true/false

Table 18: Feature call for ZTV- Zero to Ten Volt

### EXAMPLE

```
[
  {
    "featureName": "Ztv",
    "featureParameters": [
      {
        "ZtvMinDimLevel": 10,
        "ZtvCurveSelection": 1,
        "ZtvEnabled": true
      }
    ]
  }
]
```

## AmpDim

This feature is used to adjust the light output of a lamp by adjusting the mains voltage of the device.

featureName	featureParameter	Type	Remark
AmpDim		String	Case Insensitive
	VmainsStart	Integer	120V - 277V
	VmainsStop	Integer	70V - 230V
	StartDimLevel	Integer	30% - 100%
	StopDimLevel	Integer	10% - 60%

Table 19: Feature call for AmpDim

### EXAMPLE

```
[
  {
    "featureName": "AmpDim",
    "featureParameters": [
      {
        "VmainsStart": 230,
        "VmainsStop": 195,
        "StartDimLevel": 100,
        "StopDimLevel": 60
      }
    ]
  }
]
```

## MTP (Module Temperature Protection)

MTP is the method in which a thermal sensor (NTC resistor) implemented on the LED module is sensed by the driver, which will cut back output current when a predefined (temperature) limit is exceeded to protect the LED module from thermal overstress.

featureName	featureParameter	Type	Remark
MTP		String	Case Insensitive
	SelectedNtc		Case Sensitive Ntc1, Ntc2, PhilipsLedLightEngines, and Custom
	NtcBeta	Integer	Beta value between 25°C and 85°C (Beta 25/85). 2000 – 5000 Ohm
	NtcResistance	Integer	1000 – 50000 Ohm R_NTC at 25°
	DimLevel	Integer	Min dim level – 100%
	DimStart	Integer	Driver Specific
	DimStop	Integer	Driver Specific
	Enabled	Boolean	true/false

Table 20: Feature call for MTP – Module Temperature Protection

### EXAMPLE

For NTC1 (or NTC2):

```
[
  {
    "featureName": "MTP",
    "featureParameters": [
      {
        "SelectedNtc": "Ntc1",
        "DimLevel": 10,
        "DimStart": 70,
        "DimStop": 80,
        "Enabled": true
      }
    ]
  }
]
```

For Philips LED Light Engines:

```
[
  {
    "featureName": "MTP",
    "featureParameters": [
      {
        "SelectedNtc": "PhilipsLedLightEngines",
        "DimLevel": 10,
        "Enabled": true
      }
    ]
  }
]
```

For Custom:

```
[
  {
    "featureName": "MTP",
    "featureParameters": [
      {
        "SelectedNtc": "Custom",
        "NtcBeta": 3000,
        "NtcResistance": 25000,
        "DimLevel": 10,
        "DimStart": 70,
        "DimStop": 80,
        "Enabled": true
      }
    ]
  }
]
```

## Corridor

Corridor feature depends on Touch and Dim feature. The Corridor feature adjusts the light to a defined level when a presence sensor detects a person. Corridor can only be activated when the device is in Touch and Dim mode.

featureName	featureParameter	Type	Remark
Corridor		String	Case Insensitive
	NormalLevel	Integer	1-254
	BackgroundLevel	Integer	1-254
	FadeTimeIndex	Integer	0-15
	ProlongTime	Integer	0-2540
	ActivationTime	Integer	0-254
	DelayTime	Integer	0-2550
	CorridorEnabled	Boolean	true/false

Table 21: Feature call for Corridor

### EXAMPLE

```
[
  {
    "featureName": "Corridor",
    "featureParameters": [
      {
        "NormalLevel": 254,
        "BackgroundLevel": 170,
        "FadeTimeIndex": 12,
        "ProlongTime": 1800,
        "ActivationTime": 55,
        "DelayTime": 20,
        "CorridorEnabled": true
      }
    ]
  }
]
```



## Touch and Dim

This feature defines the startup behavior of a touch and dim controlled device after a power failure during dimming.

featureName	featureParameter	Type	Remark
TouchDim		String	Case Insensitive
	PowerOnLevel	Integer	Driver Specific
	IgnoreToggleLimits	Integer	true/false
	TouchDimEnabled	Boolean	true/false

Table 22: Feature call for Touch and Dim

### EXAMPLE

```
[
  {
    "featureName": "TouchDim",
    "featureParameters": [
      {
        "PowerOnLevel": 255,
        "IgnoreToggleLimits": false,
        "TouchDimEnabled": true
      }
    ]
  }
]
```

## DALI 253M

This feature provides an interface for storing information to enable predictive maintenance of a device.

featureName	featureParameter	Type	Remark
Diia253Maintenance		String	Case Insensitive
	RatedLife	Integer	kilohours
	ReferenceTemperature	Integer	Celsius
	RatedStarts	Integer	100 starts

Table 233: Feature call for DALI 253M

### EXAMPLE

```
[
  {
    "featureName": "Diia253Maintenance",
    "featureParameters": [
      {
        "RatedLife": 255,
        "ReferenceTemperature": 195,
        "RatedStarts": 65535
      }
    ]
  }
]
```

## OWP (OEM Write Protection)

OEM Write Protection (OWP) defines a password that will be set in the driver so the data of OEM Write protected features can only be written to the driver by providing the configured password.

featureName	featureParameter	Remark
Owp		Case insensitive
	OwpEnabled	true/false
		Features are case sensitive. Use the following <a href="#">CLO</a> , <a href="#">CorridorMode</a> , <a href="#">TouchDim</a> , <a href="#">MinDimLevel</a> , <a href="#">AloMin</a> , <a href="#">ALO</a> , <a href="#">DcEmergency</a> , <a href="#">AOC</a> , <a href="#">Ast</a> , <a href="#">LumInfoFormat</a> , <a href="#">LightSourceAge</a> ,
	OwpPassword	

Table 244: Feature call for OWP – OEM Write Protection

### EXAMPLE

```
[
  {
    "featureName": "Owp",
    "featureParameters": [
      {
        "OwpEnabled": [
          "Aoc",
          "LumInfoFormat"
        ],
        "OwpPassword": [
          1,
          2,
          3,
          4
        ]
      }
    ]
  }
]
```

## Multiple feature configuration with a single API call

The below syntax is an example to configure multiple features in a single API call. Please note that the example does not include all the supported features.

### EXAMPLE

```
[
  {
    "featureName": "AOC",
    "featureParameters": [
      {
        "AocEnabled": true,
        "AocCurrent": 500
      }
    ]
  },
  {
    "featureName": "CLO",
    "featureParameters": [
      {
        "CloEnabled": true,
        "CloPoint1": {
          "Time": 0,
          "Percentage": 80
        },
        "CloPoint2": {
          "Time": 100,
          "Percentage": 100
        }
      }
    ]
  },
  {
    "featureName": "DtI",
    "featureParameters": [
      {
        "DtIOutputPercentage": 50,
        "DtIDimStart": 10,
        "DtIDimStop": 88,
        "DtILampOff": 93
      }
    ]
  }
]
```

# Feature file Generation

```
Body Cookies Headers (18) Test Results 200 OK 831 ms 3.08 KB Save Response
Pretty Raw Preview Visualize XML
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <SaveData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
  XMLSchema" Version="11" ToolType="Signify MultiOne API" configurationId =
  "ad8ab4f2-f52d-410d-9f0a-39d9c9d6c4a9" TimeStamp = "2022-09-28T13:55:18.516Z" ToolVersion="9.8.3">
3 <DeviceInfo>
4   <DeviceName>Xi FP 165W 0.3-1.0A SNLDAE 230V C170 sXt</DeviceName>
5   <DeviceVersion>2.0</DeviceVersion>
6   <DeviceTwelveNc>929002875006</DeviceTwelveNc>
7 </DeviceInfo>
8 <OemWriteProtection />
9 <Features>
10 <FeatureData>
11   <FeatureName>AOC</FeatureName>
12   <FeatureVersion>1</FeatureVersion>
13   <FeatureIdExtension xsi:nil="true" />
14   <ItemKeys>
15     <string>Value</string>
```

The above figure shows an example of generated content for a feature file from Postman platform. The content can then be saved by clicking 'Save Response'. It must be saved in 'xml' file formats to be compatible with MultiOne Engineering and MultiOne Workflow. Please note that Feature files are not stored in Signify Database.

# References

HubSpot. (2022). Retrieved from HubSpot, Inc: <https://blog.hubspot.com/website/api-calls>