

# **Ki. Platform**

## **ELEXON Test Evidence**

# Table of contents

<b>Table of contents</b>	<b>2</b>
Introduction	<b>3</b>
<b>Test Evidence</b>	<b>4</b>
Test Group 1	4
Test Group 2	6
Test Group 3	10
Test Group 4	11
Test Group 5	13
Test Group 6	16
Test Group 7	16
Test Group 8	24
Test Group 9	29
Test Group 10	31
<b>Witness Test Evidence</b>	<b>33</b>
Control switch profiles	33
Control Failure - Data Assumption	34
Control Failure - Revised Data	35

# Introduction

## *Summary of Ki. and network layouts*

The Ki. Platform is a hosted CMS web application that connects to an Outdoor Device Network via LoRaWAN. The software that controls the luminaire itself is connected directly to the driver.

The software sends an entire year profile to the nodes which are executed without any further downlinks required. Lamps can be set to switch on and off based on either Lux, Realtime Clock, or Solar Clock offset.

Nodes communicate their switch on/off times via a daily billing uplink. If the CMS doesn't receive a billing uplink from a node it will not be reported in the event log and will be the responsibility of the MA to assign it with a default switch regime.

# Test Evidence

This section contains the evidence to support the requirements checklist for Central Management System (5.1 - Central Management System Equivalent Meter Test Specification)

## Test Group 1

### System Configuration

The operator of the CMS System should demonstrate the software versioning and operating platforms that will be subject to approval.

### Test Evidence

The Ki. platform source code is version controlled via Git in a private repository with monitor and secure user access control.

The screenshot displays a Git repository interface for 'project-core'. The left sidebar shows navigation options: Source, Commits, Branches, Pull requests, Pipelines, Deployments, Downloads, and Settings. The main area shows the repository's file structure and commit history. The file structure includes directories like app, bootstrap, config, database, docs, k8s, nativescript, nova, public, resources, routes, specification, src, storage, tests, and files like .codeclimate.yml, .csslintrc, .dockerignore, .editorconfig, and .env.example. The commit history table lists recent commits with their messages and timestamps. The right sidebar shows repository details, including the last update time, open pull requests, branches, watchers, forks, version control system (Git), and access level (Admin).

Name	Size	Last commit	Message
app		2 days ago	Revert "Remove ui-refresh and move everything into the main directory (pull request #61) Feature: KCP-62 - Display hover popu...
bootstrap		2018-04-25	remove call to set app name in newrelic - done automatically now
config		2019-10-28	Add config flag to disable PushData / AWS Firehose Firehose shouldn't be used in d...
database		2019-10-29	Update existing migrations to use string instead of json column type. Production da...
docs		2019-10-28	Add Site ERD to the Guidebook
k8s		2 days ago	Revert "Remove ui-refresh and move everything into the main directory (pull request #61) Feature: KCP-62 - Display hover popu...
nativescript		2019-02-18	Use node name instead of location and bump version
nova		2019-04-11	Disable local access without auth
public		yesterday	Merged in feature-kcp-62 (pull request #61) Feature: KCP-62 - Display hover popu...
resources		2 days ago	Revert "Remove ui-refresh and move everything into the main directory (pull request #61) Feature: KCP-62 - Display hover popu...
routes		2019-10-16	Add URL routes for new City Applications
specification		2017-03-03	Update DB connection details
src		2019-10-29	Remove MySQL json update code. Production runs MySQL 5.6.
storage		2019-10-28	Ignore generated Passport keys (prevents accidental committing / problems when ...
tests		2019-10-07	Merged in feature-fault-alerts (pull request #42) Feature fault alerts Approved-by: J...
.codeclimate.yml	341 B	2017-06-30	Ignore public/packages and rename eslint file
.csslintrc	107 B	2017-06-30	Add standard linting and codeclimate config
.dockerignore	4 B	2017-09-28	add kubernetes configuration
.editorconfig	374 B	2017-07-14	Update editorconfig, tidy phpunit
.env.example	678 B	2019-10-28	Add config flag to disable PushData / AWS Firehose Firehose shouldn't be used in d...

The release of new versions is handled after a build process and passing all the required automated tests. Each deployment of a new version is logged with release notes and the new release number.

#### **Ki. Platform 669 - Release Notes - 5th November 2019**

- [Release Notes - 5th November 2019](#)
- [Release Notes - 5th November 2019](#)
- [Release Notes - 5th November 2019](#)
- [Release Notes - 5th November 2019](#)

#### **Ki. Platform 647 - Release Notes - 21st October 2019**

- [Release Notes - 21st October 2019](#)

#### **Ki. Platform 635 - Release Notes - 8th October 2019**

- [Release Notes - 8th October 2019](#)

#### **Ki. Platform 634 - Release Notes - 7th October 2019**

- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)
- [Release Notes - 7th October 2019](#)

Ki. should need to resubmit for Elexon only upon a change release that includes core changes to UMSUG related data.

**Version:** ki-master-694

## Test Group 2

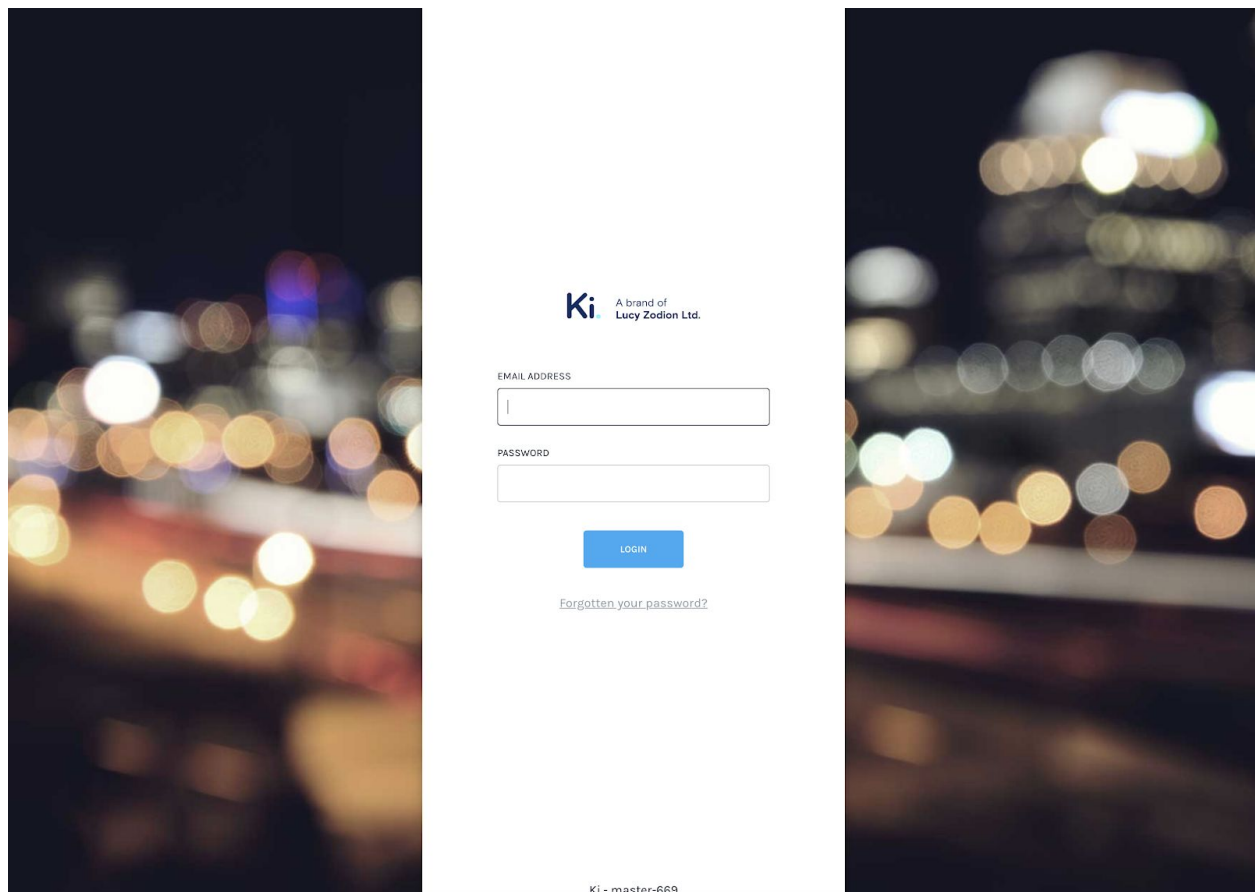
### Security

The operator of the CMS System should demonstrate the procedures which provide secure access to data. Operators should only be able to access data which is relevant to them. Secure access procedures should be demonstrated for the following participants:

- Customers

### Test Evidence

The entire Ki. platform is behind a user login system and requires valid credentials to gain access. It has security measures to avoid brute force attacks and specific encryption methods around password storage.



Users levels of access is managed via a role that is attached to the specific user. Each role contains specific permissions that are used throughout the platform to validate the current logged in user can only access what is defined.

A brand of  
Lucy Zodion Ltd.
**Roles**
Actions ▾

---

Lucy Zodion ▾

- Dashboard
- Management >
- Performance >
- Gateways >
- Users ▾
  - View All Users
  - Manage Roles
  - Manage Permissions
  - Create Host
- Development >

Name	Permissions	
Admin	Modify Permissions, Modify Gateway, Modify Node, View Gateway, View Node, Create Node, Delete Node, Modify Users, Create Gateway, Delete Gateway, View Node Group, Send Command, Debug Node, Delete Node Group, Modify Node Group, Access Application	⋮
Modifier	Modify Gateway, Modify Node, View Gateway, View Node, Send Command, Access Application	⋮
View Only	View Gateway, View Node, View Node Group, Access Application	⋮
Super Administrator	Modify Permissions, Modify Gateway, Modify Node, View Gateway, View Node, Create Node, Delete Node, Modify Users, Create Gateway, Delete Gateway, View Node Group, Send Command, Debug Node, Over-the-air Update, Delete Node Group, Modify Node Group, Switch Tenant, Create Tenant, Access Application, Access Back Office	⋮
Install Engineer	Access Commissioning App	⋮

Showing 1-5 of 5 results

Per Page: 10 ⌵

KI - master-669

- My Account
- Log Out
- Support

Permissions are split into different types of actions within different areas of the platform and connecting services / features the platform offers.

Ki

A brand of  
Lucy Zodian Ltd.

Dashboard

Management

Performance

Gateways

Users

View All Users

Manage Roles

Manage Permissions

Create Host

Development

Lucy Zodian

Ki - master-669

My Account

Log Out

Support

Lucy Zodian

▼

Information

NAME

Admin

Permissions

☒ Modify Permissions

☒ View Gateway

☒ Delete Node

☒ Delete Gateway

☒ Debug Node

☒ Modify Node Group

☐ Display EUI

☐ Access Back Office

☒ Modify Gateway

☒ View Node

☒ Modify Users

☒ View Node Group

☐ Over-the-air Update

☐ Switch Tenant

☐ Access Commissioning App

☒ Modify Node

☒ Create Node

☒ Create Gateway

☒ Send Command

☒ Delete Node Group

☐ Create Tenant

☒ Access Application

Delete Role

The platform supports attaching additional individual permissions to a specific user if required to give maximum flexibility without having to expose a permission to an entire role group.



Ki

A brand of Lucy Zodian Ltd.

Lucy Zodian

Dashboard

Management

Performance

Gateways

Users

Development

View All Users

Manage Roles

Manage Permissions

Create Host

My Account

Log Out

Support

Ki Zodian

Ki - master-669

My Account

Log Out

Support

NAME

EMAIL

PASSWORD

Disable alert emails

LANGUAGE

English

NOTES

Roles

Admin

Install Engineer

Modifier

View Only

Permissions

Modify Permissions

View Gateway

Delete Node

Delete Gateway

Debug Node

Modify Node Group

Display EUI

Access Back Office

Modify Gateway

View Node

Modify Users

View Node Group

Over-the-air Update

Switch Tenant

Access Commissioning App

Modify Node

Create Node

Create Gateway

Send Command

Delete Node Group

Create Tenant

Access Application

Delete User

Cancel

Save

Data sent from a device to the platform is handled via an API layer that a Network Provider (NP) will be configured to send data to the Ki. platform. The endpoint requires an Authorization token to be sent with the request in order for the data about a device to be stored. These are handled by Ki. and the customer directly. Any requests without a token or an invalid token will be ignored by the platform.

9

## Test Group 3

### Synchronisation to UTC

The operator of the CMS System should demonstrate that the CMS System is synchronised to UTC, either by connection to internet time servers or a radio clock, and are accurate to within  $\pm 20$  seconds per month.

### Test Evidence

The timezone is configured on an application-wide level and the default is set to UTC. This will override any server/system times at the point of application boot. All data is transformed on the intake to be stored as UTC and converted on the output to the UI to the users' actual timezone.

```
/*  
|-----  
| Application Timezone  
|-----  
|  
| Here you may specify the default timezone for your application, which  
| will be used by the PHP date and date-time functions. We have gone  
| ahead and set this to a sensible default for you out of the box.  
|  
*/  
  
'timezone' => 'UTC',
```

## Test Group 4

### **Inventory control information**

The operator of the CMS System should demonstrate the addition, modification and deletion of Inventory Control information required for the key test scenarios specified in Section 4, either manually or electronically.

The Data subject to testing are:

- Sub-Meter ID
- Effective From Data
- CMS Unit Reference
- Number of Items
- Switch Regime
- Charge Code

There is also a detailed inventory test that it is input, stored and amended correctly with an appropriate audit trail. Note this must also demonstrate where CMS operating in more than one Distribution Area, inventory information is assigned to the correct sub-meter ID. The operator of the MA System should demonstrate the recording of the audit trail for the entries made above.

### **Test Evidence**

The platform contains asset management for its own purposes but does not handle the Inventory files/spreadsheets. This would be produced and handled by the Unmetered Suppliers Operator (UMSO). The platform does handle the Sub-Meter ID, charge codes and switch regime attached to devices allowing a user to configure a device to generate the required information. The platform does have the ability to generate and export an Inventory Control information CSV document.

View

125%

Zoom

Add Category

Insert

Table

Chart

Text

Shape

Media

Comment

Collaborate

Format

Organise

Sheet 1

20191107174716

Road Reference	City	Road Name	Location	Unit Type	Unit Identity	CMS Unit Reference	Charge Code	Number of Items	Switch Regime	Number of Controls	Control Charge Code	Latitude	Longitude
Unknown		UMSUG Node 1		Street Light	117718	KI0000117718	50 0060 0002 100	1	999	1	98 0001 0022 100	53.8052800	-1.5282300
Unknown		UMSUG Node 2		Street Light	117719	KI0000117719	50 0060 0002 100	1	999	1	98 0001 0022 100	53.7874900	-1.5364800
Unknown		UMSUG Node 3		Street Light	117720	KI0000117720	50 0060 0002 100	1	999	1	98 0001 0022 100	53.8040700	-1.5321200
Unknown		UMSUG Node 4		Street Light	117721	KI0000117721	50 0060 0002 100	1	999	1	98 0001 0022 100	53.8018000	-1.5555700
Unknown		UMSUG Node 5		Street Light	117722	KI0000117722	50 0060 0002 100	1	999	1	98 0001 0022 100	53.8069200	-1.5605900
Unknown		UMSUG Node 6		Street Light	117723	KI0000117723	50 0060 0002 100	1	999	1	98 0001 0022 100	53.8036900	-1.5509700
Unknown		UMSUG Node 7		Street Light	117724	KI0000117724	50 0060 0002 100	1	999	1	98 0001 0022 100	53.7939000	-1.5592100
Unknown		UMSUG Node 8		Street Light	117725	KI0000117725	50 0060 0002 100	1	999	1	98 0001 0022 100	53.7966500	-1.5212600
Unknown		UMSUG Node 9		Street Light	117726	KI0000117726	50 0060 0002 100	1	999	1	98 0001 0022 100	53.7862800	-1.5371300
Unknown		UMSUG Node 10		Street Light	117727	KI0000117727	50 0060 0002 100	1	999	1	98 0001 0022 100	53.7944600	-1.5376000

The Sub-Meter ID can be applied on a tenant wide level or on a specific device level within the application and is used when generating the Control File and Event Log file.

Ki

A brand of  
Luvv Zedion Ltd.

ABERDEEN

Dashboard

Management

Groups

Assets

Add Asset

Profiles

Calendar

Calendar Updates

Settings

Performance

Gateways

Users

Development

Luvv Zedion

KI - master-689

My Account

Log Out

Support

Edit Asset

Cancel

Save

Asset

ASSET IDENTIFIER

UNIQUE ID

INSTALLATION DATE

ASSET TYPE

ADDRESS LINE 1

ADDRESS LINE 2

CITY / TOWN

POSTCODE

ASSET HEIGHT

SUB METER ID

DNO / PRIVATE

PHASE

ASSET NOTES

Luminaire

LUMINAIRE TYPE

DRIVER TYPE

LUMINAIRE INSTALL DATE

LUMINAIRE NOTES

DRIVER INSTALL DATE

DRIVER NOTES

Node

The audit log of any changes made to a device is stored in the revisions database with a log on who actioned the change as well as the previous value and new value.

450	App\Models\Node	168423	4	latitude	NULL	0.01443
451	App\Models\Node	168423	4	longitude	NULL	0.00937
452	App\Models\Node	168423	4	address	NULL	UMSUG Address 20
453	App\Models\Node	168423	4	fault_log	NULL	[]
454	App\Models\Node	168423	4	eui	NULL	00-00-00-20
455	App\Models\Node	168423	4	updated_at	NULL	2019-11-14 11:24:44
456	App\Models\Node	168423	4	created_at	NULL	2019-11-14 11:24:44
457	App\Models\Node	168423	4	id	NULL	168423
458	App\Models\Node	168423	4	provision_id	NULL	82KO1E
459	App\Models\Node	168423	4	tenant_id	NULL	37
460	App\Models\Node	168423	4	profile_default	NULL	{{"0":166,"1":166,"2":166,"3":166,"4":166,"5":166,"6":166}}
461	App\Models\Node	168423	4	current_calendar	NULL	[0,1,21,1,0,0,27,18,1,16,129,224,132,56,60,20,63,207,75,10,82,143,97,148,135,0,1,1,1,1,1,0,0,0]
462	App\Models\Node	168404	4	address	UMSUG Address 1	UMSUG testing node 168404
463	App\Models\Node	168405	4	address	UMSUG Address 2	UMSUG testing node 168405
464	App\Models\Node	168406	4	address	UMSUG Address 3	UMSUG testing node 168406
465	App\Models\Node	168407	4	address	UMSUG Address 4	UMSUG testing node 168407
466	App\Models\Node	168408	4	address	UMSUG Address 5	UMSUG testing node 168408
467	App\Models\Node	168409	4	address	UMSUG Address 6	UMSUG testing node 168409
468	App\Models\Node	168410	4	address	UMSUG Address 7	UMSUG testing node 168410
469	App\Models\Node	168411	4	address	UMSUG Address 8	UMSUG testing node 168411
470	App\Models\Node	168412	4	address	UMSUG Address 9	UMSUG testing node 168412
471	App\Models\Node	168413	4	address	UMSUG Address 10	UMSUG testing node 168413
472	App\Models\Node	168414	4	address	UMSUG Address 11	UMSUG testing node 168414
473	App\Models\Node	168415	4	address	UMSUG Address 12	UMSUG testing node 168415
474	App\Models\Node	168416	4	address	UMSUG Address 13	UMSUG testing node 168416
475	App\Models\Node	168417	4	address	UMSUG Address 14	UMSUG testing node 168417
476	App\Models\Node	168418	4	address	UMSUG Address 15	UMSUG testing node 168418
477	App\Models\Node	168419	4	address	UMSUG Address 16	UMSUG testing node 168419
478	App\Models\Node	168420	4	address	UMSUG Address 17	UMSUG testing node 168420
479	App\Models\Node	168421	4	address	UMSUG Address 18	UMSUG testing node 168421
480	App\Models\Node	168422	4	address	UMSUG Address 19	UMSUG testing node 168422
481	App\Models\Node	168423	4	address	UMSUG Address 20	UMSUG testing node 168423
482	App\Models\Node	168408	4	eui	00-00-00-5	00 00 00 5
483	App\Models\Node	168408	4	sub_meter_id	NULL	parall3

## Test Group 5

### Equipment control information

If applicable the operator of the CMS System should demonstrate the addition, modification and deletion of Equipment Control information required for the Scenarios described in Section 4, either manually or electronically.

The Data subject to testing are:

- CMS Unit Reference
- Sum of CMS Controller devices
- Switch Regime
- Charge Code

The operator of the MA System should demonstrate the recording of the audit trail for the entries made above.

### Test Evidence

Charge codes and regime as stored within the database and attached to devices allowing for different types of regime and charge codes to be attached to a single device with multiple components.

chargeCodes

Search

Attach ChargeCode

☐
▼

ID	COMPANY	UNIT DESCRIPTION 1	UNIT DESCRIPTION 2	CHARGE CODE	
<input type="checkbox"/> 9132	Lucy Zodion Ltd	CMS Equipment		98 0001 0022 100	
<input type="checkbox"/> 7851	Zodion	Electronic Ballast	SON HPI (MBI)	50 0060 0002 100	

Previous
1 - 2 of 2
Next

KI - v2.0.1

All changes to device data is logged via the audit logger. This will store the previous state and new state of the data along with the User ID of the who created the action. Data can be reverted to a previous state if required with the entire record being reverted or a specific field of data.

20	App\Models\Node	168423	4	address	UMSUG testing node 168423xdDAddFZ0U	UMSUG testing node 1684236ci9aZj6VL
21	App\UMSUG\ChargeCode	9148	4	unit_description_2	Control	Control Device
22	App\Models\Node	168404	4	address	UMSUG testing node 1684047m76IGi8AT	UMSUG testing node 168404AFd2IXIsOZ
23	App\Models\Node	168404	4	sub_meter_id	NULL	doD35KX
24	App\Models\Node	168405	4	address	UMSUG testing node 168405z6HBjCP9Vx	UMSUG testing node 168405Ynuw1w7GYI
25	App\Models\Node	168405	4	sub_meter_id	NULL	MsSdmIx
26	App\Models\Node	168406	4	address	UMSUG testing node 1684060hkrA5toTL	UMSUG testing node 168406UBpZEVVhvM
27	App\Models\Node	168406	4	sub_meter_id	NULL	DY4ByPp
28	App\Models\Node	168407	4	address	UMSUG testing node 168407IRJfzTFK6	UMSUG testing node 168407fL7Nshcuqn
29	App\Models\Node	168407	4	sub_meter_id	parall3	oYxnee5
30	App\Models\Node	168408	4	address	UMSUG testing node 168408FWW01Jyz1V	UMSUG testing node 168408kCpG96n0dL
31	App\Models\Node	168408	4	sub_meter_id	parall3	atb89Zt
32	App\Models\Node	168409	4	address	UMSUG testing node 168409doy3FsSFaY	UMSUG testing node 168409RGkXjsRnTJ
33	App\Models\Node	168409	4	sub_meter_id	NULL	KuKoMJo
34	App\Models\Node	168410	4	address	UMSUG testing node 1684108f77nv6MNJ	UMSUG testing node 168410GGdjxmfZLu
35	App\Models\Node	168410	4	sub_meter_id	NULL	yLJUahF
36	App\Models\Node	168411	4	address	UMSUG testing node 168411s8VN6TG2PQ	UMSUG testing node 168411sGrfQHBD7
37	App\Models\Node	168411	4	sub_meter_id	NULL	kVSa0VD
38	App\Models\Node	168412	4	address	UMSUG testing node 168412f2nef0J4mX	UMSUG testing node 168412QXs12MbW24
39	App\Models\Node	168412	4	sub_meter_id	NULL	mvQD49a
40	App\Models\Node	168413	4	address	UMSUG testing node 168413YqdkPE7KxD	UMSUG testing node 168413BLWvUwc548
41	App\Models\Node	168413	4	sub_meter_id	NULL	iHUXpmq
42	App\Models\Node	168414	4	address	UMSUG testing node 1684148ZvWDeyzFo	UMSUG testing node 168414t9QftrjWor
43	App\Models\Node	168414	4	sub_meter_id	NULL	ulkra2C
44	App\Models\Node	168415	4	address	UMSUG testing node 168415jotzEK6hX5	UMSUG testing node 168415YZo4OK9gqW

The platform has the ability to generate a Control File to be provided to UMSO to combine with other Non-CMS managed devices. The **CMS Unit Reference** is generated on output of any file that requires it. This uses a prefix of 'KI' and then is filled out for the remaining 10 characters with the internal platform device identifier.

controlparalla20191107001.log ×	
Plugins supporting *.log files found.	
1	<u>Hparalla20191107001</u>
2	KI00001177180000019995000600002100
3	KI00001177190000019995000600002100
4	KI00001177200000019995000600002100
5	KI00001177210000019995000600002100
6	KI00001177220000019995000600002100
7	KI00001177230000019995000600002100
8	KI00001177240000019995000600002100
9	KI00001177250000019995000600002100
10	KI00001177260000019995000600002100
11	KI00001177270000019995000600002100
12	Control        0000109989800010022100
13	T0000013

## Test Group 6

### **CMS Issue Instructions**

The operator of the CMS System should demonstrate the issuing of operational switching times and power level instructions for CMS Units in the CMS System for the following scenarios:

- Scenario 1 – Switch Regime 999;
- Scenario 2 – Switch Regime 998;
- Scenario 3 – Control Failure (no data for a CMS Unit);
- Scenario 4 - Revised Data after control failure (following day).

## Test Group 7

Record operational switching times and power levels

The operator of the CMS System should demonstrate the recording of operational switching times and power levels for CMS Units in the CMS System for the following scenarios:

- Scenario 1 – Switch Regime 999;
- Scenario 2 – Switch Regime 998;
- Scenario 3 – Control Failure (no data for a CMS Unit);
- Scenario 4 - Revised Data after control failure (following day)

### **Test Evidence (Combination of Test Group 6 & 7)**

Within the Ki. platform a user can create a profile which is configured on when a device should start action switch points. There are 3 types of profile triggers within Ki. These are:

- RTCC - Time-based
- Solar - Sunrise / Sunset offsets
- LUX - Light level state

Profile rules are created starting at midday and run till midday the following day. As Event Logs are per day and run midnight to midnight this may include a crossover of 2 different profiles in use which the Ki. platform will automatically handle and generate the correct profiles it should be using between the specific range.






## Switch Regime 999

A sample sunset to sunrise rule-based profile is shown below and will action the following behaviour in the given example where the sunset is 17:30 and sunrise is at 07:30. This will follow a switch regime 999 schedule that supports dimming levels.

Start	End	Percentage
1 hour before sunset (16:30)	17:00	0
17:00	20:30	80
20:30	02:30	50
02:30	Sunrise (07:30)	80
Sunrise	Next Profile Trigger	0



A brand of  
Lucy Zodion Ltd.

UMSUG

Dashboard

Management

Groups

Assets

Add Asset

Profiles

Calendar

Calendar Updates


Settings

Performance

Gateways

Users

Development



My Account

Support

Log Out

Builder

NAME

Solar Testing

COLOUR

DESCRIPTION

Save

✓

On and off points

📍

⚙️

🔄

📍

⚙️

🔄

ON SUNSET - 01:00

OFF SUNRISE + 00:00

Sunset

Sunrise

Lamp output editor

1

2

3

4

12:00

00:00

12:00

1

17:00

80 %

2

20:30

50 %

3

02:30

80 %

4

08:00

0 %

Preview

00:15

Play Simulation

## Scenario 1 - Switch Regime 999

The following example shows a profile built to the below criteria with sunset at 20:00 and sunrise at 06:00. The profile rules specify that before sunset the device is at 0% and after sunrise, it is back to 0%.

Start	End	Percentage
Sunset	20:30	100
20:30	22:00	75
22:00	23:00	50
23:00	01:00	75
01:00	Sunrise	100

UMSUG

Dashboard

Management

Groups

Assets

Add Asset

Profiles

Calendar

Calendar Updates

Settings

Performance

Gateways

Users

Development

My Account

Support

Log Out

NAME

Default

COLOUR

Save

DESCRIPTION

On and off points

Light

Settings

On/Off

Light

Settings

On/Off

12:00

00:00

12:00

ON 20:00

OFF 06:00

Preview

22:00

Play Simulation

Lamp output editor

12:00

00:00

12:00

1

20:00

100 %

2

20:30

75 %

3

22:00

50 %

4

23:00

75 %

5

01:00

100 %

6

06:00

0 %

The event log for the above profile has been generated for a specific day and showing the correct intended values based on the percentage and converted to the calculated dimming watt percentage.

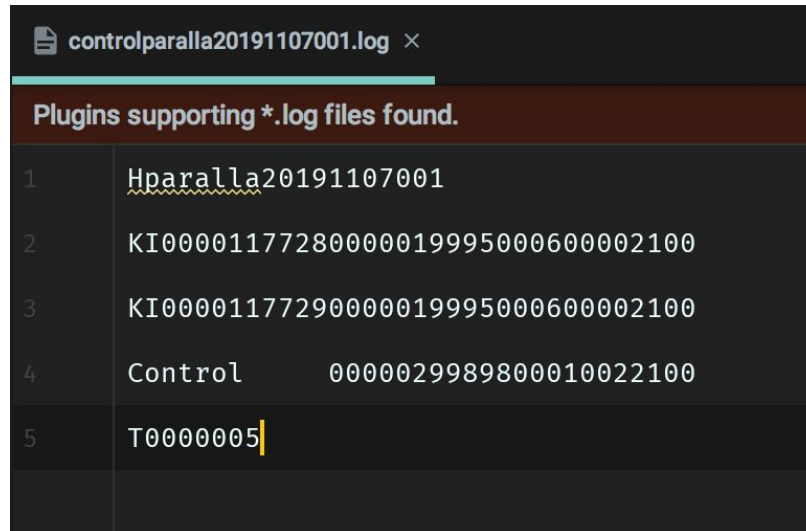
```
paralla20191106001.log x
Plugins supporting *.log files found.
1  Hparalla20191106001
2  KI0000117728000000074.000
3  KI0000117729000000050.330
4  KI0000117728010000099.000
5  KI0000117729023000079.000
6  KI0000117728060000000.000
7  KI0000117729071900000.000
8  KI0000117728200000099.000
9  KI0000117729200000099.000
10 KI0000117728203000074.000
11 KI0000117729203000074.000
12 KI0000117728220000050.330
13 KI0000117729220000050.330
14 KI0000117728230000074.000
15 KI0000117729230000074.000
16 T0000016
```

## Scenario 2 - Switch Regime 998

On the Ki. platform the test device for switch regime 998 is the KiONE Node which uses constant power consumption and sends the commands to the luminare to what output power level it should action. It is within the KiONE node that the profiles and rules are handled as the main controller.

As this type of switch regime would be continuous power is not created within the Ki. platform profile builder and is declared within the Control File it the number of devices that would be using this type and its attached charge codes. This is shown below for the 2 test records we have been counting in the number of devices against the Control group of items.

Start	End	Percentage
Continuous	Continuous	100



Plugins supporting *.log files found.	
1	Hparalla20191107001
2	KI000011772800000199950006000002100
3	KI000011772900000199950006000002100
4	Control 0000029989800010022100
5	T0000005

### Scenario 3 - Control Failure (no data for CMS Unit)

For all devices running switch regime 999 and using a profile, we will use a daily payload that tells us when profiles have been triggered on and off. This will then be used to generate the event log for the initial start points, whilst the switching points within the device profile are used at the CMS level to generate the actual levels.

There are multiple fallback methods Ki. tries to handle before not reporting a device within the event log. The Ki. platform test device does not store pending network data. Therefore if it is missed it will not be able to retry sending it at a later date. The Ki. platform will use below fallbacks based on data it does have to create event log on specific fallbacks.

**Scenario:** A device fails to send its daily payload with switch on and off times

**Fallback:** On event log generation the devices missing its daily payload will be excluded from the event log and will be assigned a default switch regime by the MA. If the device later communicates its daily payload a new version of the event log will be generated including the device previously missing.

paralla20200220001.log ×

1	<u>Hparalla20200220001</u>
2	T0000002

First generation with no switch on/off data with devices ignored.

paralla20200220002.log ×

1	<u>Hparalla20200220002</u>
2	KI0000169041123000100.000
3	KI0000169041133000078.240
4	KI0000169041140000055.630
5	KI0000169041150000078.240
6	KI0000169041160000100.000
7	KI0000169041170000004.350
8	T0000008

New version generated for the same date after having switch on/off point data with device now included.

**Scenario:** A device has a power failure or fault relating to control programs

**Fallback:** If a device has reported a fault and we know it will be at a critical level resulting in the luminaire not running a profile then it will not be shown inside the event log for the dates the fault was present.

```
paralla20191108001.log x
Plugins supporting *.log files found.
1  Hparalla20191108001
2  KI0000117728000000074.000
3  KI0000117728010000099.000
4  KI0000117728060000000.000
5  KI0000117728200000099.000
6  KI0000117728203000074.000
7  KI0000117728220000050.330
8  KI0000117728230000074.000
9  T00000009
```

Day 2 - 1 device reporting, 1 device has a power failure

Below is an audit log showing changes applied to a profile with the user responsible for the action logged. It can easily be reverted to any previous state if required.

id	revisionable_type	revisionable_id	user_id	key	old_value	new_value
12449	App\Models\Profile	160	4	off	{"type":"rtcc","value":"07:45"}	{"type":"solar","value":"+00:30"}
12448	App\Models\Profile	160	4	on	{"type":"rtcc","value":"16:15"}	{"type":"solar","value":"+02:30"}
12447	App\Models\Profile	160	4	off	{"type":"rtcc","value":"06:00"}	{"type":"rtcc","value":"07:45"}
12446	App\Models\Profile	160	4	on	{"type":"rtcc","value":"17:00"}	{"type":"rtcc","value":"16:15"}
12445	App\Models\Profile	160	4	switching_points	[{"time":"17:00","value":"80"},{...	[{"time":"17:00","value":"80"},{"time":"20:00","value":"55"},{"time":"22:15","value":"45"},{"time":"01:00","value":...
12444	App\Models\Node	168940	4	charge_code_primary_id	NULL	7851
12443	App\Models\Node	168939	4	charge_code_primary_id	NULL	7851
12442	App\Models\Node	168938	4	charge_code_primary_id	NULL	7851

## Test Group 8

### **Generate Operational Event Log - normal processing and control failure**

The operator of the CMS System should demonstrate:

The sending of a daily operational event log (CMS and MA Separate Systems) or if applicable transferring of data (CMS and MA integrated Systems) of the operational switching times and power levels for specified CMS Units to the MA in the specified format for the following scenarios:

- Scenario 1 – Switch Regime 999;
- Scenario 2 – Switch Regime 998;
- Scenario 3 – Control Failure (no data for a CMS Unit);
- Scenario 4 - Revised Data after control failure (following day)

The operator of the CMS system should demonstrate a control failure (no data for a CMS Unit) through the use of the correct information flag as per Scenario 3.

### **Operational Event Log – revision to previously reported data**

The operator of the CMS System should demonstrate that data can be revised by either issuing a refresh or an incremental operational event log (CMS and MA Separate Systems) to the MA in the specified format or if applicable the transferring of revised data (CMS and MA integrated System) from a previous control failure. (Scenario 3) The Refresh or Incremental Flow should cover:

- Refresh Flow
  - A complete refresh of the operational event logs which includes previously unknown data;
  - A complete refresh of the operational event logs which includes data which has been amended.
- Incremental Flow
  - An incremental update of operational event log which includes previously unknown data;
  - An incremental update of operational event log which includes data which has been amended.

Details of the scenarios subject to testing are given in Section 4

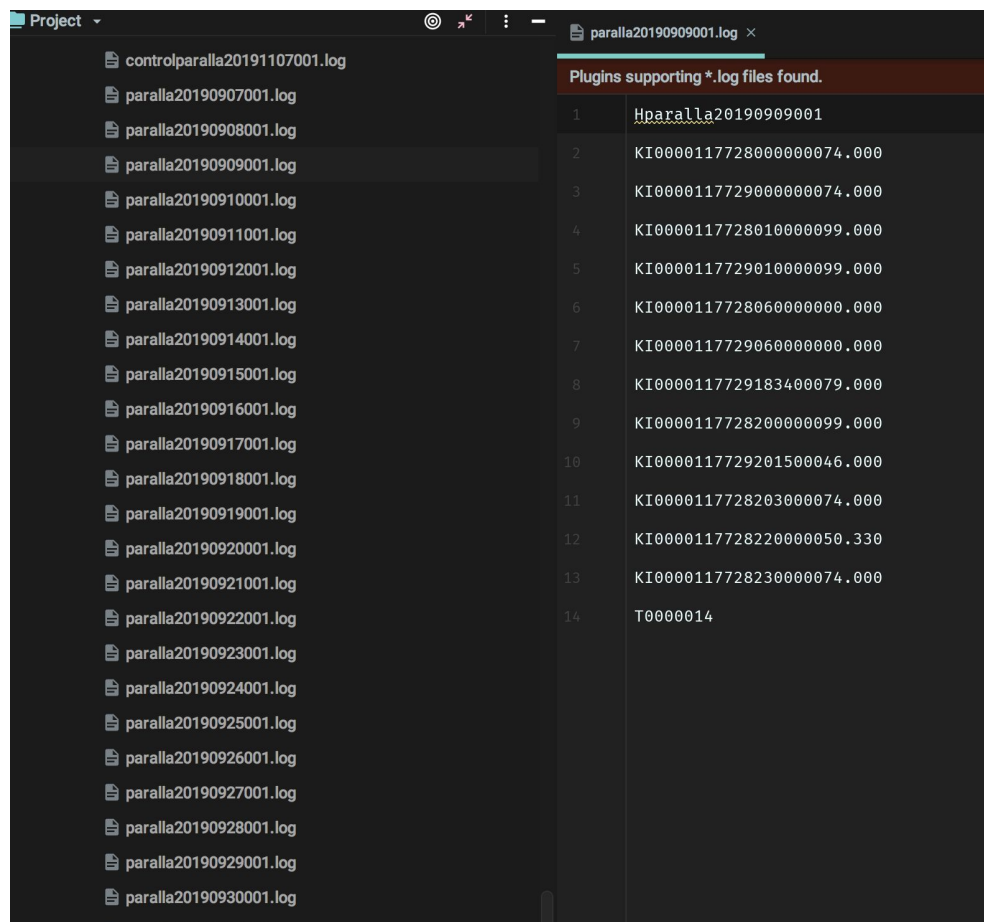
Where the CMS and MA are separate Systems the operator should demonstrate that the operational event log has been retained for audit purposes and the audit trail is correct.

Where the CMS and MA are integrated Systems the operator should demonstrate that the operational event logs can be produced for the Scenarios above for audit and testing purposes



## Test Evidence

The Ki. platform automatically generates the event log per day. However, commands exist to allow the generation of a specific date or a date range. It will go back over the data and state of devices on the given range and create event log files. As Ki. devices do not have the ability to send data after the current event time we would not have any historic data changes. Therefore Ki. will not regenerate revised event logs. However, the ability if data is manually added into the system and a specific date is asked for again it will regenerate as an incremental version as per the specification.



Below shows an event log file generated with the first version using the assumed profile the device is running with a switch-on time of 20:00 and a switch-off time of 06:00.

parall320191113001.log ×	
parall320191113002.log ×	
Plugins supporting *.log files found.	
1	Hparall320191113001
2	KI0000168439000000074.000
3	KI0000168440000000074.000
4	KI0000168439010000099.000
5	KI0000168440010000099.000
6	KI0000168439060000000.000
7	KI0000168440060000000.000
8	KI0000168439200000099.000
9	KI0000168440200000099.000
10	KI0000168439203000074.000
11	KI0000168440203000074.000
12	KI0000168439220000050.330
13	KI0000168440220000050.330
14	KI0000168439230000074.000
15	KI0000168440230000074.000
16	T0000016

The next example shows the same generation on version 2 following the Ki. platform receiving actual times for switching points with switch on at 20:05 and switch off at 05:30.

parall320191113001.log ×	
parall320191113002.log ×	
Plugins supporting *.log files found.	
1	<u>Hparall320191113002</u>
2	KI0000168439000000074.000
3	KI0000168440000000074.000
4	KI0000168439010000099.000
5	KI0000168440010000099.000
6	KI0000168439053000000.000
7	KI0000168440053000000.000
8	KI0000168439200500099.000
9	KI0000168440200500099.000
10	KI0000168439203000074.000
11	KI0000168440203000074.000
12	KI0000168439220000050.330
13	KI0000168440220000050.330
14	KI0000168439230000074.000
15	KI0000168440230000074.000
16	T0000016

All generated event log commands are recorded at the point of generation with the file path for the event log along with the following data:

- Sub Meter ID
- Filename
- Starts At (Start date of data to include in the generation)
- Ends At (End date of data to include in the generation)
- Nodes Total (Total number of devices in the file)

- Nodes Processed (Total number of actually processed nodes. This number may be less than the target due to possible faults resulting it not being included in the event log)

id	sub_meter_id	filename	starts_at	ends_at	nodes_total	nodes_processed
1	umsugts	umsug/umsug-testing/umsugts20191113004.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000
2	umsugts	umsug/umsug-testing/umsugts20191113005.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000
3	umsugts	umsug/umsug-testing/umsugts20191113006.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000
4	umsugts	umsug/umsug-testing/umsugts20191113007.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000
5	umsugts	umsug/umsug-testing/umsugts20191114001.log	2019-11-14 00:00:00	2019-11-14 23:59:59	10000	10000

Along with the file generation being recorded, each individual row that is put into the event log is recorded with the exact output line and additional useful information to verify its contents.

id	event_log_id	timestamp	row_data	event_date	event_time	node_id	percentage	watt_percentage
350000	5	1573772400	KI0000045577230000074.000	2019-11-14	23:00:00	45577	75.00	74.00
349999	5	1573768800	KI0000045577220000050.330	2019-11-14	22:00:00	45577	50.00	50.33
349998	5	1573763400	KI0000045577203000074.000	2019-11-14	20:30:00	45577	75.00	74.00
349997	5	1573761600	KI0000045577200000099.000	2019-11-14	20:00:00	45577	100.00	99.00
349996	5	1573709400	KI0000045577053000000.000	2019-11-14	05:30:00	45577	0.00	0.00
349995	5	1573693200	KI0000045577010000099.000	2019-11-14	01:00:00	45577	100.00	99.00
349994	5	1573689600	KI0000045577000000074.000	2019-11-14	00:00:00	45577	75.00	74.00
349993	5	1573772400	KI0000045576230000074.000	2019-11-14	23:00:00	45576	75.00	74.00
349992	5	1573768800	KI0000045576220000050.330	2019-11-14	22:00:00	45576	50.00	50.33
349991	5	1573763400	KI0000045576203000074.000	2019-11-14	20:30:00	45576	75.00	74.00
349990	5	1573761600	KI0000045576200000099.000	2019-11-14	20:00:00	45576	100.00	99.00
349989	5	1573709400	KI0000045576053000000.000	2019-11-14	05:30:00	45576	0.00	0.00
349988	5	1573693200	KI0000045576010000099.000	2019-11-14	01:00:00	45576	100.00	99.00
349987	5	1573689600	KI0000045576000000074.000	2019-11-14	00:00:00	45576	75.00	74.00
349986	5	1573772400	KI0000045575230000074.000	2019-11-14	23:00:00	45575	75.00	74.00
349985	5	1573768800	KI0000045575220000050.330	2019-11-14	22:00:00	45575	50.00	50.33
349984	5	1573763400	KI0000045575203000074.000	2019-11-14	20:30:00	45575	75.00	74.00
349983	5	1573761600	KI0000045575200000099.000	2019-11-14	20:00:00	45575	100.00	99.00
349982	5	1573709400	KI0000045575053000000.000	2019-11-14	05:30:00	45575	0.00	0.00

## Test Group 9

### Volume and Performance

The operator of the CMS System should provide tests evidence of volume and performance tests completed by the applicant as part of their system testing, to the accredited test agent so that compliance (within operational timescales) can be accessed.

### Test Evidence

We have created 10,000 test devices on a single Sub-Meter ID to test the performance and scalability of the event log generation. Devices have been configured with a variety of profiles with 4-8 different switch points.

The system can process 10,000 devices and generate them into a single event log file in 1 minute 2 seconds which includes checking for data available and calculating the dimming watts at the point of generation. This also includes pushing the event log file onto an FTP server to make available for the MA to fetch.

Below is a shown database log of a request to generate an entire days event log with updated\_at being the final time the record was changed after the file has been uploaded.

id	sub_meter_id	filename	starts_at	ends_at	nodes_total	nodes_processed	created_at	updated_at	mpan_id	tenant_id
1	umsugts	umsug/umsug-testing/umsugts20191113004.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000	2019-11-14 17:38:33	2019-11-14 17:39:34	umsug-testing	39
2	umsugts	umsug/umsug-testing/umsugts20191113005.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000	2019-11-14 17:40:00	2019-11-14 17:41:04	umsug-testing	39
3	umsugts	umsug/umsug-testing/umsugts20191113006.log	2019-11-13 00:00:00	2019-11-13 23:59:59	10000	10000	2019-11-14 17:41:33	2019-11-14 17:42:35	umsug-testing	39

The event log is automatically generated each day and is placed on an FTP server for the MA to fetch. We also keep all the data that is used to generate the event log and can regenerate with a simple command. Event logs can be generated for a range of dates and batch process them. Upon request the original file can be fetched and delivered, or a command can be executed to generate the physical file again based on the historic data for the defined date or date ranges.

```

Enter MPaN ID [umsug-testing]:
> umsug-mpan

Enter start date [2019-09-15]:
>

Enter end date [2019-11-14]:
>

Generating Event Log for 2019-09-15
Generating Event Log for 2019-09-16
Generating Event Log for 2019-09-17
Generating Event Log for 2019-09-18
Generating Event Log for 2019-09-19
Generating Event Log for 2019-09-20
Generating Event Log for 2019-09-21
Generating Event Log for 2019-09-22
Generating Event Log for 2019-09-23
Generating Event Log for 2019-09-24

```

id	sub_meter_id	filename	starts_at	ends_at	nodes_total	nodes_processed	created_at	updated_at	mpan_id
1	parall3	NULL	2019-09-15 00:00:00	2019-09-15 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:40	umsug-mpan
2	parall3	NULL	2019-09-16 00:00:00	2019-09-16 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:44	umsug-mpan
3	parall3	NULL	2019-09-17 00:00:00	2019-09-17 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:47	umsug-mpan
4	parall3	NULL	2019-09-18 00:00:00	2019-09-18 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:51	umsug-mpan
5	parall3	NULL	2019-09-19 00:00:00	2019-09-19 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:52	umsug-mpan
6	parall3	NULL	2019-09-20 00:00:00	2019-09-20 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:55	umsug-mpan
7	parall3	NULL	2019-09-21 00:00:00	2019-09-21 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:57	umsug-mpan
8	parall3	NULL	2019-09-22 00:00:00	2019-09-22 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:15:57	umsug-mpan
9	parall3	NULL	2019-09-23 00:00:00	2019-09-23 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:01	umsug-mpan
10	parall3	NULL	2019-09-24 00:00:00	2019-09-24 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:01	umsug-mpan
11	parall3	NULL	2019-09-25 00:00:00	2019-09-25 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:03	umsug-mpan
12	parall3	NULL	2019-09-26 00:00:00	2019-09-26 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:03	umsug-mpan
13	parall3	NULL	2019-09-27 00:00:00	2019-09-27 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:07	umsug-mpan
14	parall3	NULL	2019-09-28 00:00:00	2019-09-28 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:07	umsug-mpan
15	parall3	NULL	2019-09-29 00:00:00	2019-09-29 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:08	umsug-mpan
16	parall3	NULL	2019-09-30 00:00:00	2019-09-30 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:08	umsug-mpan
17	parall3	NULL	2019-10-01 00:00:00	2019-10-01 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:12	umsug-mpan
18	parall3	NULL	2019-10-02 00:00:00	2019-10-02 23:59:59	500	500	2019-11-15 15:15:27	2019-11-15 15:16:12	umsug-mpan



## Test Group 10

### Operational Event Log – File format

The operator of the CMS System should demonstrate that the operational event logs are in the specified format as per BSCP520 Section 4.5.2.3(c).

### Test Evidence

The event log is generated following the required format as defined below:

Filename: mmmmmmmmyyyymmddvvv.log where:  
mmmmmmmm = Sub-Meter ID (alphanumeric) yyymmdd = date to which the  
events pertain vvv = version number  
log = file extension  
with all characters in lower case

File header: HMMMMMMYYYYMMDDVVV  
where:  
H = header identifier, H  
MMMMMM = Sub-Meter ID (alphanumeric) YYYYMMDD = date to which the  
events pertain VV = version number

File body: UUUUUUUUUUHHMSSPPP.PPI  
where:  
UUUUUUUUUU = CMS Unit Reference (alphanumeric)  
HHMSS = time in hours, minutes and seconds, in UTC throughout the year  
PPP.PP = percentage of base power i.e. undimmed power level applied to the lamp, to 2  
decimal places  
I = information flag (alphanumeric)

File trailer: TNNNNNNN  
where:  
T = trailer identifier, T  
NNNNNN = total number of lines including header and trailer

paralla20191106001.log ×	
Plugins supporting *.log files found.	
1	<u>Hparalla20191106001</u>
2	KI0000117728000000074.000
3	KI0000117729000000050.330
4	KI0000117728010000099.000
5	KI0000117729023000079.000
6	KI0000117728060000000.000
7	KI0000117729071900000.000
8	KI0000117728200000099.000
9	KI0000117729200000099.000
10	KI0000117728203000074.000
11	KI0000117729203000074.000
12	KI0000117728220000050.330
13	KI0000117729220000050.330
14	KI0000117728230000074.000
15	KI0000117729230000074.000
16	T0000016

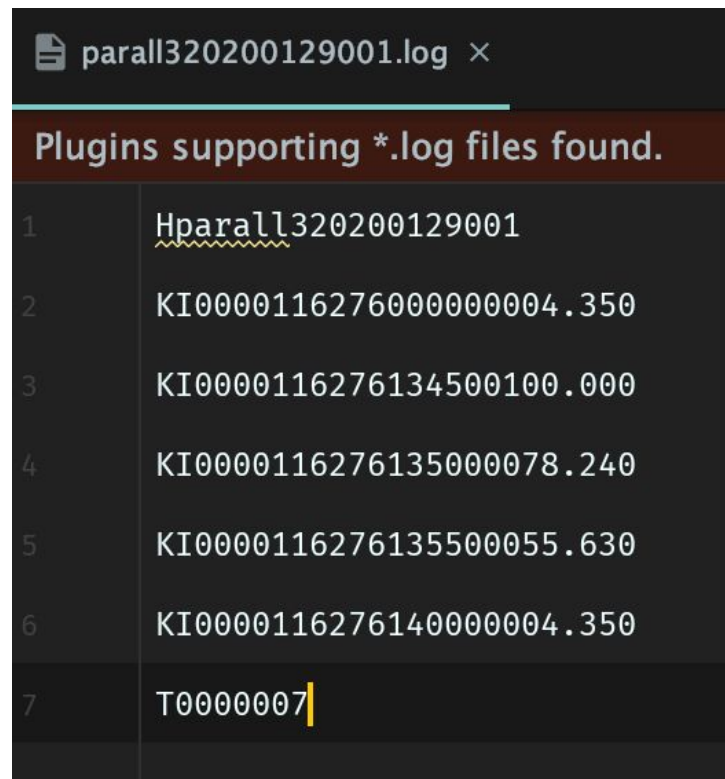


## Witness Test Evidence

This section contains event logs generated on the day during the witness test session for the given scenarios that were demonstrated.

### Control switch profiles

Generated event log showing different switch points with measure output to evidence dimming watt percentage versus brightness requests.



The screenshot shows a log file viewer window titled 'parall320200129001.log'. Below the title bar, a header row reads 'Plugins supporting \*.log files found.'. The main content is a table with 7 rows. The first row contains 'Hparall320200129001' with a yellow wavy underline. The next five rows contain 'KI' followed by a long numeric string and a decimal value. The final row contains 'T' followed by a numeric string and a yellow cursor.

Plugins supporting *.log files found.	
1	<u>Hparall320200129001</u>
2	KI00001162760000000004.350
3	KI0000116276134500100.000
4	KI0000116276135000078.240
5	KI0000116276135500055.630
6	KI0000116276140000004.350
7	T0000007

## Control Failure - Data Assumption

Revised data showing a different profile running based on assumptions without having any data from a device. Switch points are CMS driven assuming the device has run them.

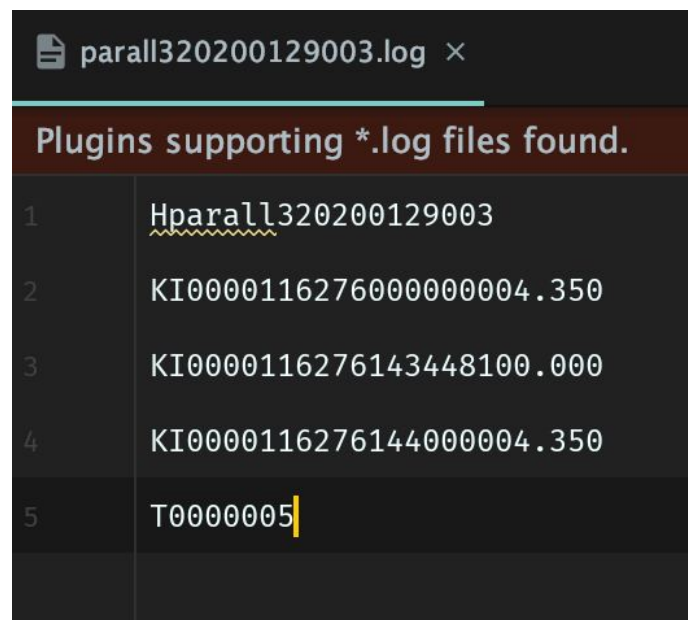
Start	End	Percentage
14:30	14:40	100
14:40	14:45	50
14:55	END	0

```
parall320200129002.log x
Plugins supporting *.log files found.
1  Hparall320200129002
2  KI00001162760000000004.350
3  KI0000116276143000100.000
4  KI0000116276144000055.630
5  KI0000116276145000004.350
6  T0000006|
```

## Control Failure - Revised Data

Evidence after device powered off during first switch point and once having actual device data that the revised output showed a different initial switch point and cut off at 14:40 as the lamp was powered off. This shows when full data is available the correct output based on the real output is generated as a valid revision.

Start	End	Percentage
14:34:48	14:40	100
14:40	END	0



The screenshot shows a log file viewer window titled "parall320200129003.log". Below the title bar, a message states "Plugins supporting \*.log files found." The log content is displayed in a list with line numbers 1 through 5. Line 1 shows "Hparall320200129003" with a yellow squiggly underline. Lines 2, 3, and 4 show "KI00001162760000000004.350", "KI0000116276143448100.000", and "KI00001162761440000004.350" respectively. Line 5 shows "T0000005" with a yellow cursor at the end. The background is dark grey.

```
parall320200129003.log ×
Plugins supporting *.log files found.
1  Hparall320200129003
2  KI00001162760000000004.350
3  KI0000116276143448100.000
4  KI00001162761440000004.350
5  T0000005
```