Inventronics: Tuner4TRONIC Cloud - Configuration API: Principles, Driver Features and Parameters

Developers' Handbook

Version 1.4

2025-08-07

Please note:
All information in this guide has been prepared with great care. INVENTRONICS, however, does not accept liability for possible errors, changes and/or omissions. Please check https://www.inventronics-light.com or contact your sales partner for an updated copy of this guide. This technical application guide is for information purposes only and aims to support you in tackling the challenges and taking full advantage of all opportunities the technology has to offer. Please note that this guide is based on own measurements, tests, specific parameters and assumptions. Individual applications may not be covered and need different handling. Responsibility and testing obligations remain with the luminaire manufacturer/OEM/application planner.

Table of Contents

Introduction

While T4T-Development provides the users an intuitive graphical interface to create luminaire production (*osrtup) files manually, the T4T Cloud REST API („Application Programming Interface“) provides software developers a collection of functions and tools to create the luminaire files automatically. Therefore, T4T Cloud REST API can be used as a replacement of T4T-Development to create the luminaire files. This speeds up the production times and minimizes human errors.

In the past, T4T knew two kind of project files: the luminaire configuration files (*.osrtul) and the read-only luminaire production files (*.osrtrup). With the introduction of the T4T CloudREST API, the T4T Cloud REST API saves the luminaire configurations automatically on your T4T Cloud account storage and now only uses one type of files: luminaire production files (*.osrtup).

Here are some examples on how you can use the T4T-Cloud REST-API:

- Create the luminaire production file automatically as soon as a new order is created in your ERP system
- Create the luminaire production file when the LED driver is going to be programmed at the assembly line
- If you have a web-tool to allow your customers to configurate and order luminaires online, T4T Cloud API can generate the luminaire production files
- If you have a mobile website or mobile app to allow your customers to download the original factory configuration of the luminaire, T4T Cloud API can generate the luminaire production files and the customers can program the files directly on the smartphone with the T4T Field app

Working with the T4T-Cloud Configuration REST-API

The T4T Cloud Configuration API is a REST API. The URL to access the Configuration API is https://www.tuner4tronic.com/t4t/api/
Clicking on the above mentioned URL, you find the list of all available API calls (get, post, etc.) as Swagger documentation.

You need an API key to authenticate. Please contact your local INVENTRONICS sales representative to obtain an API key.

In a nutshell, this is the workflow when using the T4T Clould Configuration API:

1. Create a new project
   POST /t4t/api/v1/projects → returns the ProjectID
2. Add a LED driver to the project
   POST /t4t/api/v1/drivers
3. Get the list of available Operating Modes for the selected LED driver
   GET /t4t/api/v1/drivers/{driverID}/opmodes
4. Set the desired Operating Mode
   PUT /t4t/api/v1/drivers/{driverID}/opmodes/selectbyname
5. Get the list of Feature Parameters for the selected LED driver and the selected Operating Mode
   GET /t4t/api/v1/drivers/{driverID}/properties
6. Set (update) your desired Feature Parameter values
   PATCH /t4t/api/v1/drivers/{driverID}/properties
7. Generate the production file (osrtup)
   GET /t4t/api/v1/projects/{ProjectID}/generate
   This method returns a “download id”, that can be used to download the project. If no download ID is included, there is a validation problem within the project. E.g. a parameter may be out of the range supported by the driver.
8. Download the production file (osrtup)
   GET /t4t/api/v1/projects/{ProjectID}/productionfile

In case the opsmodes and parameters are known, steps 3 & 5 can be omitted.

INVENTRONICS provides a sample project for C#/.NET to understand the basics on how to implement the T4T Cloud API in your own software.

Error Handling

When working with a Web API, proper error handling is essential for providing clear feedback to clients and ensuring reliable communication. Generally, there are two main types of errors that can occur during an API request:

1. HTTP Status Code Errors

These errors indicate that the API request has failed due to a problem on the server or with the request itself. In such cases, the API returns an appropriate HTTP status code to inform the client about the nature of the problem. Common examples include:

- **404 Not Found** – The requested resource could not be found.
- **400 Bad Request** – The request was malformed or missing required data.
- **500 Internal Server Error** – An unexpected error occurred on the server.

These error codes help clients understand why the request could not be fulfilled and take appropriate action.

2. Parameter Validation Errors

The second category of errors involves invalid parameter values. These are typically caught before the actual execution of business logic. Instead of returning an HTTP error code, the API provides a detailed ValidationSummary that outlines which parameters are invalid and why.
A ValidationSummary includes:

- isValid – is true if no errors but can have warnings
- an array of error or warning messages

Parameter validation errors can be explicitly queried with the /t4t/api/v1/projects/{id}/verify
API function or will be returned from other Api functions, like setting parameter values.

Generation and download of a production file is only possible if the current configuration does not contain any errors.

Error or warning messages are linked to the affected property or properties by the Id. To resolve the link, get all properties and do a lookup using the Id provided in the error message.

Handling of the LED module ID for a Tunable White driver

In DT8 modes the driver needs additional information about the connected LED module in order to do the color temperature calculations. The LED module ID is a reference to the LED module data in the LED Module Store. It can refer to an existing Inventronics LED module or a created custom module.
The ID can be found on the upper left side of the LED Module Store data view.

If an API is used in the Configurator, the same API key must be used for creation of custom LED modules. Otherwise, the Configurator cannot access the LED module data.

Operating Modes and Parameters

Below you find the complete list of available LED driver settings of the T4T Cloud API: from the Operating Modes to the individual LED driver feature parameters. Not all Operating Modes and Feature Parameters are available for all LED drivers; every LED driver has its own set of Operating Modes and Feature Parameters. Please have a look at the datasheet of the LED drivers you are trying to program for detailed information about their capabilities. For a detailed explanation of the individual Operating Modes and Feature Parameters please refer to the Application Guides of the LED drivers and the Tuner4TRONIC Development user manual.

Operating Modes

The T4T Cloud API offers the possibility to change the (default) Operating Mode of an LED driver. Different Operating Modes are available depending on the selected LED driver.
Note that T4T-Development uses different names for similar operating modes (e.g. depending on when the driver was released). The T4T Cloud API harmonizes these mode names to allow static mappings from external data sources to the T4T Cloud API to ease automated generation of production files. For details of the operating mode(s) see the documentation (data sheet) of the used drivers.

API Short Name	**Operating Mode Description**
Indoor Operating Modes	
dali	DALI only (single output)
dali_dual	DALI only (dual outputs)
dali_d2w	DALI with D2W output
dali8tw	DALI DT8 with Tunable White
dali_sso	DALI with Soft Switch Off
dali_sso_dual	DALI with Soft Switch Off (dual outputs)
dali_td	Autodetect: DALI or TouchDim
2dali	2 DALI channels
2dali_sso	2 DALI channels with Soft Switch Off
td	TouchDim
td_dual	TouchDim (dual outputs)
td_d2w	TouchDim with Dim2Warm output
td_2but	TouchDim with 2 buttons and 2 independent outputs
cf	Corridor Functionality
cf_dual	Corridor Functionality (dual outputs)
Outdoor Operating Modes	

astro_sd	StepDIM/ AstroDIM/ DALI (wiring selection) or AstroDIM/ LSI/ DALI (wiring selection)
astro	AstroDIM (DALI)
astro_pd	AstroDIM + Presence Detection (DALI)
astro_sdi	Autodetect: AstroDIM/ StepDIM inverse (DALI)
sd	StepDIM (DALI)
sdi	StepDIM inverse (DALI)
onoff_sdi	Autodetect: On/OFF-StepDIM inverse
md	MainsDIM (DALI)
dali	DALI only
onoff	No dimming (ON-OFF mode)
0to10	0-10 V Dimming Mode

Setting an Operating Mode

The available operating modes of a driver can be queried from the API via "**GET /t4t/api/v1/drivers/{driverId}/opmodes**". This is useful on a graphic user interface application where the user first sees the list of available operating modes and then selects the desired operating mode. In an automated application, the typical use case is that the desired operating mode is stored in a database or is generated from a 3rd party application. The desired operating mode is then set via the method "**PUT /t4t/api/v1/drivers/{driverId}/opmodes/selectbyname**".

Note that the operating modes names do not depend on the type of driver nor on a specific firmware version.

Features and their Parameters

The T4T Cloud API harmonizes the feature names (independent of the FW versions). Different features and parameters are available depending on the selected LED driver. The following table shows all parameters grouped by features and groups as they are shown in T4T-Development GUI.

The first column is the unique parameter **name**, how to call the parameter in the REST API.

The **display name** can be used to identify the parameter in the T4T-D GUI and is also available in the REST-response.

The **old API name(s)** are listed here to ease the migration of applications using the old DLL-based API, but they are no longer used in the REST API.

The **accepted values** list all values (typically) settable for the parameter and the **Unit** of that parameter. Note that for many parameters there are fixed **values** to ease deployment of INVENTRONICS drivers. However, for some parameters (e.g. current) the default and the supported range of a paramter is driver specific and thus cannot be listed in this table. Please see the technical documentation of the driver.

The next column lists the (**typical) default value**, if it is not driver specific. The API itself returns the min, max, default and off value for all parameters and the service behind checks the validity of the values set for all parameters based on the selected driver. In case the supported parameter range does not a continous list, then the API returns the acceptable values as list of values.

Parameter Name (RestAPI)	**Display Name**	**Old API Name(s)**	**Accepted Values**	**Unit**	**(Typical) Default Value**	**Comment**
AstroDIM						
Submode						
ast.mo	Submode	*Parameter was not available in the old API*	(0) Astro-based (1) Time-based (2) SD triggered	N/A	0	Available modes depend on the type of driver
Fade Timing						
ast.fs	Switch On Fade Time	AstroA-0:StartupFadeTime Astro-1:StartupFadeTime Astro-2:StartupFadeTime	0, 15, 30, 45, 60, 120, 300, 450, 600, 750, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3600	s	0	
ast.fd	AstroDIM Fade Time	AstroA-0:AstroDIMFadeTime Astro-1:AstroDIMFadeTime Astro-2:AstroDIMFadeTime	0, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 90, 120, 180, 240, 300, 360, 480	s	180	
ast.fe	Switch Off Fade Time	AstroA-0:SwitchOFFFadeTime Astro-1:SwitchOFFFadeTime Astro-2:SwitchOFFFadeTime	0, 15, 30, 45, 60, 120, 300, 450, 600, 750, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3600, Off	s	Off	
Reference Schedule						
ast.l1	Output Level 1	AstroA-0:NominalLevel Astro-1:NominalLevel Astro-2:NominalLevel	0-100	%	100	

ast.l2	Output Level 2	AstroA-0:DimLevel1, Astro-1:DimLevel1, Astro-2:DimLevel1	0-100	%	50	
ast.l3	Output Level 3	AstroA-0:DimLevel2, Astro-1:DimLevel2, Astro-2:DimLevel2	0-100	%	50	
ast.l4	Output Level 4	AstroA-0:DimLevel3, Astro-1:DimLevel3, Astro-2:DimLevel3	0-100	%	50	
ast.l5	Output Level 5	AstroA-0:DimLevel4, Astro-1:DimLevel4, Astro-2:DimLevel4	0-100	%	100	
ast.t0	Time 2	AstroA-0:DimStartTime, Astro-1:DimStartTime, Astro-2:DimStartTime	00:00 ... 23:59	hh:mm	22:00	
ast.t1	Time 3	AstroA-0:DimDuration1, Astro-1:DimDuration1, Astro-2:DimDuration1	00:00 ... 23:59	hh:mm	00:00	
ast.t2	Time 4	AstroA-0:DimDuration2, Astro-1:DimDuration2, Astro-2:DimDuration2	00:00 ... 23:59	hh:mm	02:00	
ast.t3	Time 5	AstroA-0:DimDuration3, Astro-1:DimDuration3, Astro-2:DimDuration3	00:00 ... 23:59	hh:mm	04:00	
Location						
	Latitude	AstroA-0:Latitude, Astro-1:Latitude, Astro-2:Latitude	-90 ... 90	° resolution: 0.125°	0	Use the PUT location method of the REST API to the set the common location for all drivers in the project.
	Longitude	AstroA-0:Longitude, Astro-1:Longitude, Astro-2:Longitude	-180 ...180	° resolution: 0.125°	0	
	Time Zone	AstroA-0:UTCTimeShift, Astro-1:UTCTimeShift, Astro-2:UTCTimeShift	-12 ... 14	h to UTC resolution: 0.25h	0	
Constant Lumen Output						
clm.en	Enable	ConstLum-0:Command-ClmEnable	Enabled (1) or Disabled (0)	N/A	(0)	Enable (1) activates the Constant Lumen feature
clm.l1	Output Level 1	ConstLum-0:AdjustmentLevel1	70-100	%	driver specific	most drivers only support 70..100%
clm.l2	Output Level 2	ConstLum-0:AdjustmentLevel2	70-100	%	driver specific	
clm.l3	Output Level 3	ConstLum-0:AdjustmentLevel3	70-100	%	driver specific	
clm.l4	Output Level 4	ConstLum-0:AdjustmentLevel4	70-100	%	driver specific	
clm.l5	Output Level 5	ConstLum-0:AdjustmentLevel5	70-100	%	driver specific	
clm.l6	Output Level 6	ConstLum-0:AdjustmentLevel6	70-100	%	driver specific	
clm.l7	Output Level 7	ConstLum-0:AdjustmentLevel7	70-100	%	driver specific	

clm.l8	Output Level 8	ConstLum-0:AdjustmentLevel8	70-100	%	driver specific	
clm.t1	Operating Time 1	ConstLum-0:Time1	0	kh	0	fixed value, must be 0 kh
clm.t2	Operating Time 2	ConstLum-0:Time2	1 ... 254	kh	driver specific	
clm.t3	Operating Time 3	ConstLum-0:Time3	2 ... 254	kh	driver specific	
clm.t4	Operating Time 4	ConstLum-0:Time4	3 ... 254	kh	driver specific	
clm.t5	Operating Time 5	ConstLum-0:Time5	4 ... 254	kh	driver specific	
clm.t6	Operating Time 6	ConstLum-0:Time6	5 ... 254	kh	driver specific	
clm.t7	Operating Time 7	ConstLum-0:Time7	6 ... 254	kh	driver specific	
clm.t8	Operating Time 8	ConstLum-0:Time8	7 ... 254	kh	driver specific	
Operating Current						
cur.en	Enable		Enabled (1)=Fixed Current Disabled (0)=LEDSet Mode	N/A	depends on the type of driver	Enabled (1) means that the "Fixed Current Mode" becomes active. Disable (0) means that the driver automatically determines the current setting, e.g. via a LEDSet resistor.
cur.0	Current	OTConfig-0:NominalLEDCurrent OTConfig-1:DefaultOpCurrent OTConfig-2:DefaultOpCurrent PwmConfig-0:DefaultOpCurrent OTConfig-3:DefaultOpCurrent	depends on the type of driver e.g. 200 - 1050	mA	depends on the type of driver	
cur.X	Current		depends on the type of driver	mA	depends on the type of driver	X={0,1,2,3,...} In case of drivers with N outputs, the cur.X property exists N times. One case are TW drivers where Output 0 is uses for Cold White and Output 1 is used for the Warm White LEDs.
Dim to Dark						
d2d.en	Enable	D2D-0:Config-Enable GFM-1:D2DConfig-Enable	Enabled (1), Disabled (0)	N/A	Disabled	
DALI / DEXAL Settings						
Dimming Curve						
da.dc	DimmingCurve	DimmingCurve	Logarithmic (0), Linear (1)	N/A	Logarithmic	

Fade						
da.ft	Fade Time	FadeTime	0.0, **X,** 0.7, 1.0, 1.4, 2.0, 2.8, 4.0, 5.7, 8.0, 11.3, 16.0, 22.6, 32.0, 45.3, 64.0, 90.5	s	0	This parameter "includes" the fast fade times **X** in case that feature in available in the driver. The fast fade times have a range of 0.050 to 0.675 s, depending on the driver capability.
da.ef	Extended Fadetime	ExtendedFadeTime	0.1 to 960	s	1	In T4T-D the extended fade time was set by setting Base and Multiplier separately. In the API, this replaced by setting the effective fade time duration in s. As only the (Standard) Fade Time or the Extended Fade Time can be active in a driver, the user can control that by setting the other fade time to zero. When both fade times are set, the driver will use the FadeTime (as specified in the DALI standard).
da.fr	Fade Rate	FadeRate	2.8, 4.0, 5.6, 7.9, 11.2, 15.8, 22.4, 31.6, 44.7, 63.3, 89.4, 127, 179, 253, 358	steps/s	44.7	
Limits						
da.ma	Max Level	MaxLevel	1-254	DALI Level	254	Max level can't be lower than Min level
da.mi	Min Level	MinLevel	1-254	DALI Level	85	Min level can't be higher than Max level. Level should be greater than or equal to physical Min level (depends on LED driver)

Power ON						
da.po	Power On Level	PowerOnLevel	0-254, Off	DALI Level	254	Off (MASK) = Enable "Use last stored level on Power ON" Power on level should be in the range between DALI Min and Max levels.
d8.poc	Power On Colour Temperature	PowerOnColour	1000-10000	K	4000	Colour temperature must be within the physical colour temperature range.
System Failure						
da.sf	System Failure Level	SystemFailureLevel	0-254, Off	DALI Level	254	Off (MASK) = Enable "Don't change existing level on System Failure" System failure level should be in the range between DALI Min and Max levels
d8.sfc	System Failure Colour Temperature	SysFailureColour	1000-10000	K	4000	Colour temperature must be within the physical colour temperature range.
DALI DT8-TW Settings						
Colour Temperature Range						
d8.cc	Coldest Colour Temperature	CtCoolest	1000 - 10000	K	5000	
d8.wc	Warmest Colour Temperature	CtWarmest	1000 - 10000	K	3003	
LED Module Configuration						
d8.pcc	Physical Coldest Colour Temperature	CtPhysCoolest	2625 - 10000	K	6494	
d8.pwc	Physical Warmest Colour Temperature	CtPhysWarmest	1000 - 6452	K	2703	

Driver Guard						
dg.en	Enable		Enabled (1), Disabled (0)	N/A	Disabled	
dg.dl	DeratingLevel	DG-0:DeratingLevel	Output Level	%	50	This parameter is the dimlevel reached during temperature derating. For most drivers the level is fixed and cannot be modified.
dg.pre	Prestart Derating	DG-0:PrestartDerating	0, -5, -10, -15, -20	°C	0	This is temperature before the Td max, when the derating starts. Using this feature, the derating starts at a lower temperature.
dg.pd	Power Derating			W		available only in specific drivers
Dimming Modes						
dm.en	Enable	OTConfig-3:PWM-Threshold	Analog Dimming (0), Hybrid Dimming (1)	N/A	depends on type of driver	Disabled (0) means "Analog Dimming" Enable (1) means "Hybrid" (PWM dimming above driver specific threshold
DEXAL Power Supply Unit						
dx.en	Enable	DexalPSU-0:Enable	Enabled (1) Disabled (0)	N/A	depends on type of driver	available only in specific drivers
Emergency Mode						
em.en	Enable		Enabled (1) Disabled (0)	N/A	depends on type of driver	
em.set	Level		0-254	DALI level	depends on type of driver	
em.ld	Lock DALI Parameter	Emergency-0:ConfigLock-DaliParamLock	Enabled (1) Disabled (0)	N/A	Disabled	
End of Life						
eol.en	Enable	EOL-0:Command-EOLEnable	Enabled (1) Disabled (0)	N/A	Disabled	
eol.ti	Life Time	EOL-0:EOLTime	0-254	kh	50	

Luminaire Info*						
inf.ver	Content Format ID	Bank1:ContentFormatVersion	2, 3	N/A	empty	The version number that shall be set into the Content Format Version field. Type 2 is the INVENTRONICS propietary format for certain LED drivers (e.g. OT 1DIM/4DIM) and Type 3 is the standardized parameter set according to the DiiA specification and available in DEXAL INVENTRONICS LED drivers.
General						
inf.oemg	Luminaire Manufacturer GTIN	Bank1:OEMGTIN	6 byte integer	N/A	empty	Parameter independent of Content Format. Always present when Luminiare Info feature is available.
inf.oemid	Luminaire Identification number	Bank1:OEMIdentification	6 byte integer	N/A	empty	Parameter independent of Content Format. Always present when Luminiare Info feature is available.
inf.con	Content	Bank1:VendorSpecificContent	block of "hexbytes"; the lenght depends on the Content Format Version, for type 3 it is 101 bytes.	N/A	empty	The complete bock of Luminiare Info data. The mapping of individual luminaire info parameters to the position is within the block is in the reponsibility of the API user. Or instaed of changing this property use the LumInfos API functions to access Luminaire info properties.
Parameter						
inf.ebl	Expected Ballast LifeTime	Bank207:ExpectedBallastLifeTime	0-254	kh	empty	Optional parameter, only available with luminaire Info Type 3.
inf.idt	Internal Driver Reference Temperature	Bank207:IntDriverRefTemp	0-194	°C	empty	Optional parameter, onlyavailable with luminaire Info Type 3.
inf.elt	Expected LightSource LifeTime	Bank207:ExpectedLightSourceLifeTime	0 - 65535	kcycle	empty	Optional parameter, only available with luminaire Info Type 3.
OEM Key & Configuration Lock*						
OEM Key						
key.en	Enable			N/A		
key.all	Full or Partial Protection		All (1), Partial (0)	N/A		Only available in drivers with OEM Key: 1 indicates that all protectable features are protected by the OEM key. 0 indicates that only the safety relevant parameters are protected (called "partial protection")
Configuration Lock						
key.mp	Master Key	MSK-0:NewMasterPwd PwmConfig-0:MasterPwd		N/A		
key.sp	Service Key	MSK-0:ServicePwd PwmConfig-0:ServicePwd		N/A		
Access to Features						
key.ps	Permission Service	MSK-0:PermService PwmConfig-0:PermService		N/A		

key.pu	Permission User	MSK-0:PermUser PwmConfig-0:PermUser		N/A		
Lamp Operating Time						
ops.en	Enable		Enabled (1), Disabled (0)	N/A	1	Cannot be disabled on most drivers
ops.ti	Time	Info-0:LampOperationCounter	0-279	kh	0	
SoftSwitch Off						
sso.en	Enable	GFM-0:SSOMode-Enable GFM-1:SSOMode-Enable	Enabled (1), Disabled (0)	N/A	Disabled	
sso.ti	Time	GFM-0:SSOMode-Time GFM-1:SSOMode-Time	0.7, 1.0, 1.4, 2.0, 2.8, 4.0, 5.7	s	0.7	

TouchDIM+Sensor						
Button 1: Light Level Control						
td.mo	TouchDim Mode		1, 2, 3	N/A		This parameter is Read-Only. The touchdim mode is set via the operating mode. This parameter is only a shortcut to inform the API user, in which mode the driver is an which parameters are valid. 1 = basic TouchDim mode 2 = 2 independent outputs (2 button) 3 = Tunable White output
Button 1: Light Level Control						
td.es1	Enable Short-Push (Switch On/Off)	TDiCorridor-0:TDConfig-ShortPushEnable TDiCorridor-1:TDConfig-ShortPushEnable	Enabled, Disabled	N/A	Enabled	
td.el1	Enable Long-Push (Dim Up/Down)	TDiCorridor-0:TDConfig-LongPushEnable TDiCorridor-1:TDConfig-LongPushEnable	Enabled, Disabled	N/A	Enabled	
td.ed1	Enable Double-Push (Set/Reset Level)	TDiCorridor-0:TDConfig-DoublPushEnable TDiCorridor-1:TDConfig-DoublPushEnable	Enabled, Disabled	N/A	Enabled	
Button 2: Light Level Control (in 2 indepentent output mode) or Colour Temperature Control (in Tunable White mode)						Tunable White LED drivers only with Operating Mode TouchDIM Tunable White
td.es2	Enable Short-Push (TW: Set Colour to middle)		Enabled, Disabled	N/A	Enabled	Mode 2 and 3 only
td.el2	Enable Long-Push (TW: Colour WARMER/COLDER)		Enabled, Disabled	N/A	Enabled	Mode 2 and 3 only
td.ed2	Enable Double-Push (TW: Set/Reset Colour)		Enabled, Disabled	N/A	Enabled	Mode 2 and 3 only
Power ON settings						
td.po	Mains Power On Level	TDiCorridor-0:TDPowerOnLevel TDiCorridor-1:TDPowerOnLevel	0-255	DALI level or 255	255	255 = "Start with active level before last Power Off"
td.po2	Main Power On Level Output 2	N/A	0-255	DALI level or 255		255 = "Start with active level before last Power Off" Mode 2 only

td.poc	Mains Power ON Colour		1000-10000, Off	K	Start with active colour temperature before last Power Off	Off= "Start with active colour temperature before last Power Off" The colour temperature should be between the physical coldest and warmest colour temperature Mode 3 only	
Operative settings							
td.fu	Fade (Up) Time [1]	TDiCorridor-0:FadeUpTime TDiCorridor-1:FadeUpTime	selected values between 0 ... 90.5	s	0.7	This parameter is shared between TD and CF and can only set to one common value	
td.so	TD Switch on Level	TDiCorridor-0:TDSwitchOnLevel TDiCorridor-1:TDSwitchOnLevel	1-255	DALI Level or 255	255	255 = "Use active level before last Switch Off".	
td.so2	TD Switch on Level Output 2	N/A	1-255	DALI Level or 255	255	255 = "Use active level before last Switch Off" Mode 2 only	
td.soc	TD Switch on Colour		1000 - 10000, Off	K	4000	Off= "Start with active colour temperature before last Switch Off" The colour temperature should be between the physical coldest and warmest colour temperature Mode 3 only	
td.on	Presence Hold Time [2]	TDiCorridor-0:PSTimeout TDiCorridor-1:PSTimeout	selected values between 5 ... 1305 (=21 min 45s) and ∞ (infinite)	s	900s = 15min		
td.fs	Fade (Down) Time [3]	TDiCorridor-0:CFFadeTime2 TDiCorridor-1:CFFadeTime2	selected values between 0 - 90.5	s	0	This parameter is shared between TD and CF and can only set to one common value	
Standby							
td.sb	Fade Time [4]	TDiCorridor-0:FadeDownTime TDiCorridor-1:FadeDownTime	selected values between 0 - 90.5	s	00:32.000	Fade Time (3) for TouchDIM is the same as Fade Time (4) for TouchDIM Sensor	
N/A	TD StandBy Level	N/A	0 - 254	DALI Level	85	The StandBy Level must be set via the DALI Min Level (da.mi); there is no separate parameter for it.	
td.st	Hold Time [5]	TDiCorridor-0:TDStandByTime TDiCorridor-1:TDStandByTime	selected values between 5 ... 1305 (=21 min 45s) and ∞ (infinite)	s	5		

Sensors						
td.els	Enable Light Sensor	TDiCorridor-0:TDConfig-LSEnable TDiCorridor-1:TDConfig-LSEnable	Enabled, Disabled	N/A	Enabled	
td.epd	Enable Presence Detection	TDiCorridor-0:TDConfig-PSEnable TDiCorridor-1:TDConfig-PSEnable	Enabled, Disabled	N/A	Enabled	
td.ehm	Enable Holiday Mode	TDiCorridor-0:TDConfig-HolidayEnable TDiCorridor-1:TDConfig-HolidayEnable	Enabled, Disabled	N/A	Enabled	
td.eap	Auto Disable Presence Detection	TDiCorridor-0:TDConfig-AutoDisPSEnable TDiCorridor-1:TDConfig-AutoDisPSEnable	Enabled, Disabled	N/A	Disabled	
Corridor						
Power ON settings						
td.su	Corridor Power On Behavior	N/A	0: standard power on behaiour 1: use L1 at power on 2: use L2 at power on 255: stay off	N/A	0	Not available on all drivers
Operative						
td.fu	Fade Up Time [1]	TDiCorridor-0:FadeUpTime TDiCorridor-1:FadeUpTime	selected values between 0 ... 90.5	s	0.7s	This parameter is shared between TD and CF and can only set to one common value
td.l0	Corridor Operative Level	TDiCorridor-0:CFLevel0 TDiCorridor-1:CFLevel0	1 - 254	DALI Level	254	
td.h0	CF Hold Time 0 [2]	TDiCorridor-0:CFHoldTime0 TDiCorridor-1:CFHoldTime0	selected values between 0, 30, ... 7758 (=2h 7min 38s) and ∞ (infinite)	s	151 (=2min 31s)	
Standby 1						
td.fd	Fade Down Time [3]	TDiCorridor-0:FadeDownTime TDiCorridor-1:FadeDownTime	selected values between 0 - 90.5	s	32s	This parameter is shared between TD and CF and can only set to one common value
td.l1	Corridor Standby Level 1	TDiCorridor-0:CFLevel1 TDiCorridor-1:CFLevel1	0 - 254	DALI Level	170	
td.h1	CF Hold Time 1 [4]	TDiCorridor-0:CFHoldTime1 TDiCorridor-1:CFHoldTime1	selected values between 0, 30, ... 7758 (=2h 7min 38s) and ∞ (infinite)	s	∞	

Standby 2						Standby2 can be used only if StandBy1 Hold Time [4] is not set to OFF or Infinite
td.fs	CF Fade Down Time 1-2 [5]	TDiCorridor-0:CFFadeTime2 TDiCorridor-1:CFFadeTime2	selected values between 0 - 90.5	s	level not active as previous hold time is infinite	This parameter is shared between TD and CF and can only set to one common value
td.l2	Corridor Standby Level 2	TDiCorridor-0:CFLevel2 TDiCorridor-1:CFLevel2	0 - 254	DALI level	level not active as previous hold time is infinite	
td.h2	CF Hold Time 2 [6]	TDiCorridor-0:CFHoldTime2 TDiCorridor-1:CFHoldTime2	selected values between 0, 30, ... 7758 (= 2h 7min 38s) and ∞ (infinite)	s	level not active as previous hold time is infinite	
Tuning Factor						
tf.en	Enable	TF-0:Enable	Enabled (1) Disabled (0)	N/A	Disabled	
Limits and Reference Lumen						
tf.ma	Max Limit	TF-0:MaximumTuningFactor	50-100	%	100	
tf.mi	Min Limit	TF-0:MinimumTuningFactor	50-100	%	50	
tf.re	Reference	TF-0:ReferenceLumenOutput	0 - 65000, Off	Lumen	Off	Luminaire reference light output at 100% Dim Level
Tuning Level						
tf.le	Level	OTConfig-1:TuningFactor TF-0:TuningFactor	50-100	%	100	must be within Min TF and Max FT Limits
StepDIM/LSI						
Output Levels						
sd.l0	Nominal Level		0-100	%	100	
sd.l1	StepDIM Level		0-100	%	50	
Switch on Timing						
sd.f0	Switch on Fade Time		0, 15, 30, 45, 60, 120, 300, 450, 600, 750, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3600	s	0	
StepDIM Timing						
sd.f1	Start Fade Time		0, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 90, 120, 180, 240, 300, 360, 480	s	180	
sd.h1	Hold Time		0, 15, 30, 45, 60, 120, 300, 450, 600, 750, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3600	s	0	
sd.f2	End Fade Time		0, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 90, 120, 180, 240, 300, 360, 480	s	180	

MainsDIM						
md.sv	Start Voltage		190-250	V	220	Start Voltage must be 30V higher than Stop Voltage
md.sl	Start Level		31-100	%	100	Start Level must be higher than Stop Level
md.ev	Stop Voltage		170-200	V	180	
md.el	Stop Level		30-85	%	30	
Thermal Protection						
tp.en	Enable		Disabled (0) Enabled (1)	N/A	driver specific	
tp.mo	Mode		Resistor Based (0) or Temperature Based (1)	N/A	driver specific	Temperature-based mode is not available all drivers
tp.se	Sensor Type		NCP18XH103J (0) NCP18XW153J,(1) NCP15XW153E+390, (2) EPCOS B57423V2473H (3)	N/A	EPCOS B57423V2473H	available only if Temperature Based mode is selected
Settings						
tp.sr	Start Derating Resistance		driver specific, typically 5000-25000 Ohm	Ohm	6300 Ohm	used in resistor-based mode
tp.er	End Derating Resistance		driver specific, typically 5000-25000 Ohm	Ohm	5000 Ohm	used in resistor-based mode
tp.or	Shut Off Resistance		driver specific, typically 5000-25000 Ohm or Off	Ohm	4500 Ohm or Off	used in resistor-based mode: Off means no final shut off. Not supported in all drivers.
tp.st	Start Derating Temperature		driver specific, typically 60 - 93°C	°C		alternative to tp.sr when temperature based mode is used
tp.et	End Derating Temperature		driver specific, typically 60 - 93°C	°C		alternative to tp.er when temperature based mode is used
tp.ot	Shut Off Temperature		driver specific, typically 60 - 93°C	°C		alternative to tp.or when temperature based mode is used
tp.dl	Derating Level		driver specific, typically 10-100%	%	50%	
Presence Detection						
pd.l	PD Level		0-100	%	100	
pd.sf	Start Fade Time		0, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 90, 120, 180, 240, 300, 360, 480	s	0	
pd.h	Hold Time		0, 15, 30, 45, 60, 120, 300, 450, 600, 750, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3600	s	0	
pd.ef	End Fade Time		0, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 90, 120, 180, 240, 300, 360, 480	s	4	
LED Module Data						
led.id	Select existing LED Module			N/A		ID of the TW LED Module
LED Module Operating Data						
led.tc	Module temperature Cold White (CH 1)		0 - 125	°C		

led.tw	Module temperature Warm White (CH 2)		0 - 125	°C		
led.ta	Module ambient temperature		0 -100	°C		
led.mo	LED module mode		0 = Basic 1 = Advanced 2 = Premium	N/A		read-only: mode of the selected LED module
led.cf	Cold White - Luminous Flux		0...65535	lm		read-only: flux at set operating current
led.wf	Warm White - Luminous Flux		0...65535	lm		read-only: flux at set operating current
led.cx	Cold White - Color Coordinate x		0..1			read-only: color at set operating current
led.cy	Cold White - Color Coordinate y		0..1			read-only: color at set operating current
led.wx	Warm White - Color Coordinate y		0..1			read-only: color at set operating current
led.wy	Warm White - Color Coordinate y		0..1			read-only: color at set operating current
led.ct	Cold White - Color Temperature		1000 - 10000	K		read-only: color at set operating current
led.wt	Warm White - Color Temperature		1000 - 10000	K		read-only: color at set operating current
Dim to Warm						
dw.c1	Colour Temperature 1		1000 - 10000, off	K	6494	
dw.c1	Colour Temperature 2		1000 - 10000, off	K	2703	
dw.c2	Colour Temperature 3		1000 - 10000	K	off	
dw.c3	Colour Temperature 4		1000 - 10000	K	off	
dw.l0	Dimming Level 1		0 - 100	%	100	
dw.l1	Dimming Level 2		0 - 100	%	0	
dw.l2	Dimming Level 3		0 - 100, off	%	off	
dw.l3	Dimming Level 4		0 - 100, off	%	off	

Notes

- **TouchDim / Corridor**
  For the TD/CF feature, the INVENTRONICS driver base the timing for hold times on the number of waves on the power line, in order to precisely synchronize longer times between multiple drivers/luminaires in a room, so that there is no visual difference when using the same settings. As the number of waves per second is different for the US (60Hz) and Europe (50Hz), the resulting hold times are different, depending on where the driver is deployed. This frequency setting is not programmed into the driver and thus not "compensated" by the driver. Instead the "different resulting behavior" has to be known and compensated in the configuration tool.
  E.g. if the hold time is 60s for 50Hz system, then the same hold time is 60s*50/60 = 50s in a 60 Hz system.
  The API gives the list of hold time values per parameter based on a 50Hz system. Your application needs to convert those values when the driver is operated in a 60Hz environment by 5/6.

- **Configuration Lock**

WARNING: We strongly advise to set up a password protection for your luminaire configurations to avoid unintentional or unauthorized adjustment of crucial paramenters that might lead to damage or non-safe operation of the luminaire.
Please also consider the required luminaire certifications and the effect of parameter adjustments on the validity of such certifications.

- **Luminaire Info**

In the ***inf.con*** property, the complete block of Luminiare Info data can be implemented.

Here is an example how to set the parameter "Nominal Input Power":
According to the DiiA Specification "DALI Part 251 – Memory bank 1 extension" that parameter is a 2 byte value located at address 0x15 (=21). The Luminaire Info (Type 3) has 101 bytes for the data block and the start address of the data block is 0x13 (=19). Then, that parameter has to be set as follows:

```
content = bytes[101]                    # create an object of 101 bytes, i.e. proper length for Luminaire Info type 3
content [(21 – 19)] = bytes("0x012C")   # user wants to set value to 300W = 0x12C
```

Then the object "content" must be submitted to the API as the parameter inf.con.
Note: The REST API does not support binary data, therefore the parameter inf.con has to be encoded as hexbytes.

As an alternative the provided /t4t/api/v1/luminfos API functions can be used. They will decode / encode the luminfo properties inside the inf.con property.

Setting Multiple Parameters

There is a shortcut to set one or more parameters by their name when the parameter names (short REST API names) are known. The method is called "**PATCH /t4t/api/v1/drivers/{driverId}/namedproperties**". The parameter short names do not depend on the type of driver nor on the firmware version. For example, the following block sets the operating current into fixed current mode and selects a current of 311 mA and disables the Constant Lumen feature:

```
[{ "name":"cur.en", "value":"1"},

{ "name":"cur.0",  "value":"311"},

 { "name":"clm.en", "value":"0"}, ]
```

In this way, the API user can set all parameters of a driver in one shot.

Some (sub) features can be disabled or tuned to a specific behaviour. This can be done by setting the property to the specific value "Off" (case sensitive). For example the AstroDim Switch Off Fade Time (ast.fe). For the DALI Power On Level (da.po) "Off" means the power down value is saved and used as power on value. In the DALI standard this special value is called **MASK**.

Get the List of Supported Values for a Parameter

The **method "GET /t4t/api/v1/drivers/{driverId}/properties"** returns a list of properties. Depending on the property the value of the property can have different formats, additional informations about the format is given in the ValueMetaInfo.

- Numeric
  The valid value range is given in the Min and Max of the ValueMetaInfo.
- Hexbytes
  Is used for binary content and formatted as hex string. Example "FF3B12AA". The length in bytes is given in the length of the ValueMetaInfo.
- Enum
  A discrete list of available values. As the the available values for a given property also depending on the driver capabilities, the property contains a complete list of available values in the availableValues of the ValueMetaInfo.
  When changing an enum property, the provided new value **must** be one of the possible values from the ValueMetaInfo.availableValues
  For example, the accepted values for the DALI fade time are "0, 0.7, 1, 1.4, 2, 2.8, 4, 5.7, 8, 11.3, 16, 22.6, 32, 45.3, 64, 90.5". The API user can simply set as the desired value="5.7" instead of determining an index for that value or do a conversion.

Some Common Used Enum Property Values

Important: Depending on the driver capabilities and different DALI standard version when the driver was created the available set can differ. Its always recomended to use the provided set of available values from the property.

Extended Fadetime (da.ef):
0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1; 1.1; 1.2; 1.3; 1.4; 1.5; 1.6; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 20; 30; 40; 50; 60; 70; 80; 90; 100; 110; 120; 130; 140; 150; 160; 180; 240; 300; 360; 420; 480; 540; 600; 660; 720; 780; 840; 900; 960

TD Fade Time / CF Fade Time (td.fu, td.fd, td.fs)
0; 0.7; 1; 1.4; 2; 2.8; 4; 5.7; 8; 11.3; 16; 22.6; 32; 45.3; 64; 90.5; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.8; 0.9; 1.1; 1.2; 1.3; 1.5; 1.6

TD Hold Time (td.on, td.sb)
5; 10; 15; 20; 25; 30; 35; 40; 46; 51; 56; 61; 66; 71; 76; 81; 87; 92; 97; 102; 107; 112; 117; 122; 127; 133; 138; 143; 148; 153; 158; 163; 168; 174; 179; 184; 189; 194; 199; 204; 209; 215; 220; 225; 230; 235; 240; 245; 250; 255; 261; 266; 271; 276; 281; 286; 291; 296; 302; 307; 312; 317; 322; 327; 332; 337; 343; 348; 353; 358; 363; 368; 373; 378; 383; 389; 394; 399; 404; 409; 414; 419; 424; 430; 435; 440; 445; 450; 455; 460; 465; 471; 476; 481; 486; 491; 496; 501; 506; 511; 517; 522; 527; 532; 537; 542; 547; 552; 558; 563; 568; 573; 578; 583; 588; 593; 599; 604; 609; 614; 619; 624; 629; 634; 639; 645; 650; 655; 660; 665; 670; 675; 680; 686; 691; 696; 701; 706; 711; 716; 721; 727; 732; 737; 742; 747; 752; 757; 762; 767; 773; 778; 783; 788; 793; 798; 803; 808; 814; 819; 824; 829; 834; 839; 844; 849; 855; 860; 865; 870; 875; 880; 885; 890; 895; 901; 906; 911; 916; 921; 926; 931; 936; 942; 947; 952; 957; 962; 967; 972; 977; 983; 988; 993; 998; 1003; 1008; 1013; 1018; 1023; 1029; 1034; 1039; 1044; 1049; 1054; 1059; 1064; 1070; 1075; 1080; 1085; 1090; 1095; 1100; 1105; 1111; 1116; 1121; 1126; 1131; 1136; 1141; 1146; 1151; 1157; 1162; 1167; 1172; 1177; 1182; 1187; 1192; 1198; 1203; 1208; 1213; 1218; 1223; 1228; 1233; 1239; 1244; 1249; 1254; 1259; 1264; 1269; 1274; 1279; 1285; 1290; 1295; 1300; 1305; Infinite

CF Hold Time (td.h1, td.h2)

0; 30; 60; 90; 120; 151; 181; 211; 241; 271; 301; 332; 362; 392; 422; 452; 482; 512; 543; 573; 603; 633; 663; 693; 724; 754; 784; 814; 844; 874; 904; 935; 965; 995; 1025; 1055; 1085; 1115; 1146; 1176; 1206; 1236; 1266; 1296; 1327; 1357; 1387; 1417; 1447; 1477; 1507; 1538; 1568; 1598; 1628; 1658; 1688; 1719; 1749; 1779; 1809; 1839; 1869; 1899; 1930; 1960; 1990; 2020; 2050; 2080; 2110; 2141; 2171; 2201; 2231; 2261; 2291; 2322; 2352; 2382; 2412; 2442; 2472; 2502; 2533; 2563; 2593; 2623; 2653; 2683; 2714; 2744; 2774; 2804; 2834; 2864; 2894; 2925; 2955; 2985; 3015; 3045; 3075; 3106; 3136; 3166; 3196; 3226; 3256; 3286; 3317; 3347; 3377; 3407; 3437; 3467; 3497; 3528; 3558; 3588; 3618; 3648; 3678; 3709; 3739; 3769; 3799; 3829; 3859; 3889; 3920; 3950; 3980; 4010; 4040; 4070; 4101; 4131; 4161; 4191; 4221; 4251; 4281; 4312; 4342; 4372; 4402; 4432; 4462; 4492; 4523; 4553; 4583; 4613; 4643; 4673; 4704; 4734; 4764; 4794; 4824; 4854; 4884; 4915; 4945; 4975; 5005; 5035; 5065; 5096; 5126; 5156; 5186; 5216; 5246; 5276; 5307; 5337; 5367; 5397; 5427; 5457; 5487; 5518; 5548; 5578; 5608; 5638; 5668; 5699; 5729; 5759; 5789; 5819; 5849; 5879; 5910; 5940; 5970; 6000; 6030; 6060; 6091; 6121; 6151; 6181; 6211; 6241; 6271; 6302; 6332; 6362; 6392; 6422; 6452; 6482; 6513; 6543; 6573; 6603; 6633; 6663; 6694; 6724; 6754; 6784; 6814; 6844; 6874; 6905; 6935; 6965; 6995; 7025; 7055; 7086; 7116; 7146; 7176; 7206; 7236; 7266; 7297; 7327; 7357; 7387; 7417; 7447; 7477; 7508; 7538; 7568; 7598; 7628; 7658; Infinite