



## DOMOSAPIENS ADVANCED LIGHTING SELECTOR

This driver is used to **dynamically select** and **activate/deactivate Advanced Lighting Scenes**, based on simple or elaborate Control Logic you define in Properties. When requested (triggered) by the user, Scene selection may be controlled by time of day as well as the state of several conditions (other lights or scenes, blinds, security partitions, contacts, day of week, ambient lighting, and programming variables). This is intended to be very flexible and powerful, yet easy to specify.

In addition, with its **extensive ALS Reports**, this driver may be used to **document** all elements of Advanced Lighting Scenes in a Project (something that Composer does not yet provide), as well as produce a comprehensive Cross Reference list of Scenes, Lights and Rooms.

---

### QUICK START – A SIMPLE EXAMPLE

**Ask:** the client wants to use one keypad button to activate/deactivate a Lighting Scene during the day and a different Lighting Scene at night. The keypad button LED must show the state of the Scene.

#### **Solution:**

- Install the driver. Do not modify any properties other than the ones outlined below. This example uses **Simple** capabilities.
- In Agents/Advanced Lighting, create the Daytime and the Nighttime Lighting Scenes to turn on the required light(s) at the required level(s). Also create the Default Toggle Scene for each. As you have modified the Scenes setup, run the driver's Action '*Refresh Advanced Lighting Scenes*'.
- Use the properties *Select Advanced Lighting Scene A* to point to your Daytime Scene and *Select Advanced Lighting Scene B* to point to your Nighttime Scene.
- Set the *Control Logic When Activating/ON* property to: ***d = A ; n = B*** This means that the first program will activate Scene **A** during daytime (lowercase **d**) and the second program will activate Scene **B** during nighttime (lowercase **n**).
- Connect the driver's *Toggle Button Link L* to the desired keypad button. You may modify the properties for *Keypad LED Active/ON Color* and *Keypad LED Inactive/OFF Color* if you wish.

You are done! Now, read on...

---

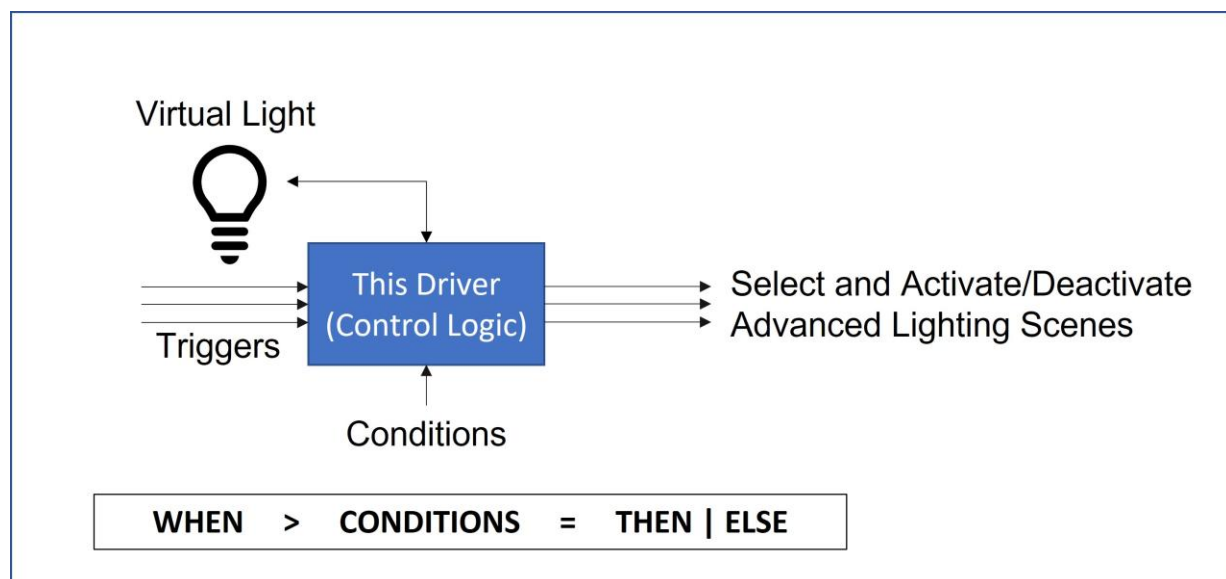
### OVERALL DESCRIPTION AND CAPABILITIES (READ THIS!)

The use of this driver can be very simple and yet quite sophisticated, allowing for a very large number of scenarios, often difficult and even impossible to do via programming. It is recommended that you read this

document completely to have a better feel for what is possible and how to do it. The many examples provided should help you get a head start.

You would typically use many copies of this driver in your Project, one for each instance where a Lighting Scene needs to be selected following a user action (such as pressing a keypad button, for example). You might very well have a 6-button keypad where each button is connected to its own instance of this driver (often via the driver's Toggle connection, which is the most flexible).

**(\*NEW\*)** The driver has three Capability levels: **Simple**, **Medium** and **Full**. If you are new to the driver, start with **Simple**. This will be enough to select Advanced Lighting Scenes from a keypad button and optionally based on time of day. Several properties will be hidden to simplify your use of the driver. **Medium** gives you additional capabilities for controlling ON and OFF states. **Full** opens up the Conditional section as well as the ELSE clauses. The sections below assume **Full** capabilities. If using **Simple** or **Medium**, simply skip over the properties you do not see.



The driver implements a '**WHEN > CONDITIONS = THEN | ELSE**' model. In other words, the driver is usually triggered by a user or system action, such as the user pressing a keypad button, a contact being activated (a door being opened, for example), the driver's own Virtual light being turned ON or a programming command being issued. This is the **WHEN** part.

The **main property** used by the driver is the *Control Logic When Activating/ON* property where you instruct the driver on what to do when it is triggered ON. This property contains up to 20 different programs (separated by semicolons) and each program follows the **WHEN > CONDITION = THEN | ELSE** model. There are two types of programs: programs with a time element and programs without a time element. The two types may be intermixed. For example, you may want a keypad button (or a door contact) to trigger different scenes depending on the time of day but also always trigger some scenes no matter what time it is. The programs containing a time element are combined to form a full day's schedule where only one of these programs would be allowed to perform its action depending on the current time of day. This is further explained in the section **CONTROL LOGIC SPECIFICATION** below.

The driver uses letters to refer to logic elements and letters **A** to **J** (ten in all) refer to Selected Advanced Lighting Scenes which are defined in the driver's properties. In its simplest use, the driver may toggle

several Scenes at the press of a keypad button (no conditions). To manage four scenes, for example, the Control Logic could simply read: **ABCD**. In longhand, this would be equivalent to:  **$L > = A+B+C+D+$**  (the trigger is the keypad/Virtual Light L, no conditions between > and =, and the four scenes **A** to **D** are activated +). Spaces are always optional and may be used liberally to improve legibility.

**(\*NEW\*) (Medium and Full ONLY)** You may specify different logic for Activating/ON and Deactivating/OFF as you may have specific scene layouts in the Project for ON and OFF. For example, you may have different scenes you wish to Activate depending on the time of day, but always use the same scene(s) to Deactivate/OFF. In addition, a special **Alternate** Control Logic is provided to allow the user to Activate an alternate set of scenes. A use case for this would be to normally select scenes which have different light levels based on time of day when the user presses a keypad button, but to allow for a scene with Full ON, for example, when the user taps the Alternate ON keypad button or, when not connected, double taps the keypad ON button.

**(Full ONLY)** The Conditional logic is also quite powerful, but all elements are optional and may be combined in many ways, including 'and' and 'or'. For example, you could Activate Scene **A** when the door opens but not when the blind is down, where Scene **B** would be Activated instead. Except when the Security System is ARMED (Scene **C** could turn ON) or when the lights in the next room are already ON (nothing happens). And on... This should enable most scenarios to satisfy your client's creativity.

**(\*NEW\*) (Full ONLY)** In addition to letters **A** to **J** used to refer to Selected Scenes in the Control Logic, the letter **K** refers to the Triggering Contact and **L** to the Virtual Light (hence also to the keypad buttons). The letters **M** to **Q** (five in all) refer to Conditional devices, **R** refers to a Conditional room, and **S** to **V** (four in all), to Conditional Contacts. **W** is used for Weekday conditions, **X**, **x**, **Y** and **y** are four Conditional variables of different types you may set and then test as part of the Control Logic and **Z** is the Ambient Light Conditional (when connected to the *Domosapiens Ambient Light Manager* driver, available separately on DriverCentral)

The driver provides **utility Output Relays and Contacts** which are activated when their corresponding Advanced Lighting Scene is active. **EXTRA FEATURE:** the driver will Activate or Deactivate an Advanced Lighting Scene if its corresponding Relay is operated externally by another process in the Project. This is independent of the driver's triggering mechanism.

In addition, the driver has a specific Action to **Print ALS Reports**. Experiment with it, as these reports give you a view as well as a capability to document, **not currently available in Composer**.

---

## INSTALLATION

1. Use the free trial period or activate the driver at any time by assigning to this project the license you purchased from the DriverCentral website (requires the DriverCentral cloud driver). You may use multiple copies of this driver in your project under the same license.
2. Select the *Driver Capabilities* level. Properties will be hidden or shown as appropriate.
3. First select the *Number of Advanced Lighting Scenes* you wish to use (up to 10) and select each Scene (**A** to **J**) in the appropriate property. You would often want to consider only Scenes Having a Toggle Scene, but this is not mandatory.

4. The driver can Activate or Deactivate one **Output Contact** per Selected Scene. If you wish to use these, connect appropriate contacts, and specify the normal (deactivated) state for All or an individual normal (deactivated) state for each.
5. The driver may be triggered to select and Activate/Deactivate a Scene (as per the Control Logic) by a **Triggering Contact (K)**. If you wish to use this facility, connect a contact, and specify its normal (deactivated) state.
6. The driver may also be triggered by up to three **keypad** buttons (ON, OFF and Toggle). If you wish to use this, connect the appropriate button(s) and specify the Keypad LED colors.
7. The driver may also be triggered by its own **Virtual Light** being turned ON or OFF, by an Advanced Lighting Scene, for example. Simply add its Load to desired scene(s). Remember that Selected Scenes in the driver may **NOT** contain the driver's own Virtual Light, as this could cause a loop.
8. **(\*NEW\*) (Full ONLY)** The Control Logic to select which Scene(s) to Activate/Deactivate when triggered, may contain **conditions**. These may be the states of Lights or Blinds or Security Partitions in the Project. Up to five of such conditional devices may be selected and referred to as **M** to **Q**. Similarly, a room (referred to as **R**) and up to four conditional contacts may be used in the control logic (referred to as **S** to **V**). Connect such contacts and specify their normal (deactivated) states as required.
9. Once these steps are completed, you get to specify the actual **Control Logic**. See the section **CONTROL LOGIC SPECIFICATION** below for syntax, details, and examples. You should at least specify the *Control Logic when Activating/ON*. The other two Control Logics are optional. Also refer to the section **OVERALL DESCRIPTION AND CAPABILITIES** above for more information.
10. The driver provides several reports to help you **document the Advanced Lighting Scenes** in your Project. Use as appropriate. The driver also provides a report to list the Control Logics you have specified, as a possible aid to troubleshooting.
11. When you are done, remember to Refresh Navigators.

---

## OVERALL DRIVER PROCESS (READ THIS!)

The basic premise of this driver is that the user wishes to turn something ON or OFF, directly or indirectly. The driver can thus be triggered to act using one of several ways:

1. A keypad button (ON, OFF or Toggle) connected to the driver (holding to ramp up/down is supported).
2. **(\*NEW\*)** A Tap on the keypad Alternate ON button or, when not connected, a Double Tap on the keypad ON or Toggle buttons, but only if the *Control Logic When Alternate Activating/ON* property is coded.
3. The driver's Virtual Light being turned ON or OFF by something in the Project, including another Advanced Lighting Scene.
4. The driver's Triggering Contact being activated or deactivated.

5. A programming command being issued to the driver: *Trigger Control Logic as 'p'*.
6. The installer running the driver's Action *Trigger Control Logic as 'p'*.

**(Medium or Full ONLY)** When the driver receives a trigger, it first determines if this is an ON, an Alternate ON or an OFF trigger. For the first two, the driver refers to the corresponding Control Logic property and executes the one or several programs it contains. If this is an OFF, the driver looks at the *Behavior When Deactivating/OFF* property and determines the action to take, which may include using the *Control Logic When Deactivating/OFF* property and executing the one or several programs it contains.

If any one of the Control Logic statements (ON, Alternate or OFF) is to be executed, the driver then determines the statement's timetable by looking at each program in the statement. As timing is optional, refer to the discussion in the section **CONTROL LOGIC SPECIFICATION** below. Then, each program in the statement (the programs are separated by semicolons) is examined for the following criteria:

1. Is the program authorized to respond to this trigger? This is done by comparing the actual trigger with what is specified in each program before the '>' character. This specification is entirely optional and by default (*KLp>*), each program responds to all triggers.
2. Does the program have a time condition? This is optional and, if specified, it must be the first condition. Programs with a time condition will only run if the current time matches it and no other program replaces it with a later time. Use the Action *Print Decoded Control Logic* to review the timing elements for all programs.
3. Does the program have any other conditions? These are evaluated one by one for each program and combined with any specified **&** ('and' - the default) and **|** ('or') operators into a final answer (true or false) for the program. Each program is independent in this regard.
4. Once the driver has determined that the program's conditions are satisfied (the result is true), or when no conditions have been specified, as they are optional, the program's actions are executed, and Scenes are Activated (+) or Deactivated (-) as requested.

When using the Keypad buttons for example, the ON button would activate the *Control Logic When Activating/ON* as would the Toggle button if the Virtual Light is currently Off. This ON action may end up Activating some Scenes and Deactivating others (with a – sign). At the end, the Virtual Light will be set to ON if any scene has been activated by the operation. Otherwise, the Virtual Light will be set to OFF.

**NOTE:** it is possible that complex scenarios could cause the driver to set a wrong state for the Virtual Light. You are encouraged to experiment to find alternative Control Logic / Lighting Scene specifications or even to open a ticket at DriverCentral to document your challenge. We might even consider enhancing the driver based on such challenges, if it is possible and makes sense to do so.

---

## CONTROL LOGIC SPECIFICATION

**Note:** this applies to the *Control Logic When Activating/ON*, the *Control Logic When Alternate Activating/ON* and *Control Logic When Deactivating/OFF* properties. It is important to note that these three properties are entirely independent of each other. This discussion then pertains to any Control Logic

statement. Spaces may be used liberally for legibility (as they are, in our Examples) and are ignored by the driver.

The driver implements a **WHEN > CONDITIONS = THEN | ELSE** model and this is specified by the Control Logic statements. View a statement as a series of 'programs'. From one to twenty programs may be specified in each statement, programs are separated by a semicolon (;). For example, **d=AB ; n=DC-** has two simple programs: during daytime (lowercase d), triggering the driver (by pressing the keypad ON button) activates Scenes **A** and **B** and during nighttime (lowercase n), triggering the driver activates Scene **D** and deactivates Scene **C**. The + sign is optional and is the default. The – sign, however, must always be specified.

Programs may be used to specify actions at different times of the day, or they may have no time specified which means they will run at any time. The two types may be mixed. Although many formats are possible, an **easy way** to view this is as a full day's schedule starting at 0:00 hours (this driver uses a 24h time representation), with the program being valid from its specified time until the next specified time in another program or until midnight. The **exception** to this is when using **n** (nighttime) as the latest time in the day, where the program will be available until sunrise the next day. Use the Action *Print Decoded Control Logic* to analyze the logic and see the times when each program is available. You do not need to specify programs in ascending time order, as the driver will sort this out.

Example:

**d = A+ ; n = D+ ; 2300 = ED- ; F**

Using four programs: if the driver is triggered during daytime (sunrise to sunset), it activates Scene **A**; if the driver is triggered during nighttime (after sunset), but before 11pm, it activates Scene **D** (the + is optional). From 11pm until midnight, the driver would activate Scene **E** and deactivate Scene **D**. At all times during the day, triggering the driver will also activate Scene **F** (no conditions). The only program which spills unto the next day is **n** used as the latest time of the day, which is not the case here. To have the 2300 program available until sunrise the following day, you would specify (notice the full day schedule):

**0000 = ED- ; d = A+ ; n = D+ ; 2300 = ED- ; F**

**IMPORTANT:** each Control Logic statement is independent, and the times listed in its programs apply only to it and not to the other two Control Logic statements.

**(\*NEW\*)** Each program element (separated by semicolons) uses one of the following formats:

**Triggers > Conditions = Actions** (**Full ONLY**, with **Triggers >** and **Conditions =** optional)

**Time Condition = Actions** (**All capabilities**, executed if time condition is met)

or simply

**Actions** (**All capabilities**, always executed)

**Description of Triggers , Conditions and Actions:**

- **Triggers > (Full ONLY)** is optional and allows specific triggers to be used for this program only. The default is **KLP>**. The three letters may be specified in any order and the > must be omitted if no letters are specified (the default is used), which is what you would normally want.
  - **K** indicates that the program may be triggered by the Triggering Contact
  - **L** indicates that the program may be triggered by the Virtual Light/Keypad
  - **p** (lowercase) indicates that the program may be triggered by a programming command

Examples with a trigger specification (in the fourth program):

**d = A+ ; n = D+ ; 2300 = ED- ; Lp> F**

The first three programs are run by any trigger (they use the default **KLP>**), but the fourth program will NOT run when the driver is triggered by the Triggering Contact (**K**), which is missing from the specification.

- **Conditions = (Full ONLY, except for Time)** is also optional but perhaps the most powerful part of this driver. The default is no conditions, so the specific program (see **Actions**) is free to run. Conditions may be specified as described below, in any order, except for the time condition (**d**, **n** or **HHMM**) which must always be first when present. Each condition element (except the time condition) is either true or false and the elements are connected by **&** (meaning 'and', the default) or by **|** (meaning 'or'). The **&** operator has higher precedence over the **|** operator and is thus evaluated first. Parenthesis may be used freely to modify or ensure the order of evaluation.

For example: **MOP- = A** is equivalent to **( M+ & O+ & P- ) = A+** which means that Scene **A** will be activated only if devices **M** and **O** are ON and device **P** is OFF. Depending on the type of device selected for these, the + and - signs may have different meanings (see below).

The following: **MO | PR = A** is equivalent to **( M+ & O+ ) | ( P+ & R+ ) = A+** which means that Scene **A** will be activated either if device **M** is ON as well as device **O**, or if device **P** is ON and contact **R** is activated. Combinations and logic are really limited by your imagination.

A time condition may be added (with **&**, the default, or even with **|**) in front of the other conditions: **0600 & M- = A+** which means that Scene **A** will be activated after 6am but only when device **M** is OFF (for example, **M** could be a blind where **M-** means the blind is DOWN/CLOSED - see below). Similarly, **0600 | M- = A+** would activate Scene **A** after 6am or when device **M** is OFF. As described earlier, this program is valid only until the next program with a time condition starts.

The other important point to consider is that the conditions are all evaluated at the same time, when the driver is triggered (by pressing a keypad button, for example). Therefore, when Selected Scenes are used as conditions, any action which changes them will not modify the conditions.

The following conditions are thus available:

- Time conditions: **d** (lowercase) for daytime, **n** (lowercase) for nighttime and **HHMM** for other times. In addition, **d** and **n** may have a + or – offset of up to 90 minutes. For example, **d+30** is valid starting 30 minutes after sunrise and **n-60** is valid starting one hour before sunset.
- **A to J**: a Selected Lighting Scene must be in the specified state for the condition to be true. The state may be: + (Active) or – (Inactive).

- **M to Q:** a device must be in the specified state for the condition to be true. This state depends on the type of device (+ is always the default):
    - If the device is a Light, the state may be: ++ (fully ON at 100%), + (ON at any level), - or - - (OFF).
    - If the device is a Blind, the state may be: ++ (fully UP/OPEN), + (OPEN at any level), - or - - (fully DOWN/CLOSED).
    - If the device is a Security Partition, the state may be: ++ (ARMED AWAY), + (ARMED any type), - or - - (DISARMED).
  - **(\*NEW\*) R:** the selected room must be in the specified state for the condition to be true. The state may be: + (room Power is ON) or – (room Power is OFF).
  - **S to V:** a Conditional Contact must be in the specified state for the condition to be true. The state may be + (activated) or – (deactivated).
  - **W:** specifies which day(s) of the week the condition is true (+) or false (-). The format is **W** followed by square brackets containing days separated by commas with ranges indicated by a **(\*NEW\*)** colon (:) such as: **W [Su, Mo, Tu, Fr, Sa]** which is equivalent to **W+ [Su:Tu, Fr:Sa]**. If you specify a minus sign in front of the square brackets, then the days listed are when the condition will be false. For example, **W- [Su]** means the condition is true on all days except Sundays. This is equivalent to **W+ [Mo:Sa]**.
  - **(\*NEW\*) X and x (lowercase):** specify the numerical value(s) which the variable **X** or the variable **x** (the two variables are distinct) may take (+) or not take (-) for the condition to be true. Ranges are allowed (with colons :) and values must be separated with commas within the square brackets. Examples: **X+ [22, 33, 44, 55]** which means that the condition will be true if variable **X** contains one of the four listed values. The following is also possible: **X+ [22:37, 55, 100:125]** which means the condition will be true if variable **X** contains one of the following values: 22 to 37, 55 and 100 to 125. Similarly, the condition **X- [55:77]** will be false if variable **X** contains a value between 55 and 77 (true otherwise).
- (\*NEW\*)** These two variables may be negative and also have a decimal part. For example, **x+ [-50.2 : 40.8]** will be true if **VARIABLE\_NUM\_x** contains any number between -50.2 and 40.8 (such as -12 or 27.9995, for example). **IMPORTANT NOTE:** if using a number with a decimal, always use a decimal point here, even if your country uses another symbol for the decimal part (such as a comma, for example). When setting the variable in Programming, you would use your country's decimal symbol, as the driver will process this properly.
- **Y:** specifies the string values which the variable **Y** may take (+) or not take (-) for the condition to be true. Ranges are NOT allowed here. Examples: **Y+ [HOME, WORK, SLEEP]** which means that the condition will be true if variable **Y** has one of the three values listed. Similarly, **Y- [WORK, VACATION]** means the condition will be true only if variable **Y** has any value other than the two values listed. To avoid issues, do NOT use the following characters as part of variable **Y** values: **square brackets [ ], parenthesis ( ), greater than symbol >, equal sign =, commas and spaces.**



- **(\*NEW\*) y (lowercase)** specifies the value of this Boolean variable which will cause the condition to be true. **y+** will be true if the variable contains the value 'True' and false otherwise. Similarly **y-** will be true if the variable contains the value 'False' and false otherwise.
- **Z:** a selected instance of the Domosapiens Ambient Light Manager driver (available separately on DriverCentral) must be in the specified state for the condition to be true. The states are **++** (BRIGHT), **+** (BRIGHT or MEDIUM), **-** (MEDIUM OR DARK), **--** (DARK).
- **Actions** (are a mandatory element, as they designate the Selected Scene(s) which will be Activated (+, the default) or Deactivated (-) if the conditions are met or if no conditions are specified. Simply list the Selected Scenes and use + and – as appropriate. For example: **ABC** (equivalent to **A+B+C+**) activates Scenes **A**, **B** and **C**. Similarly, **A+B-C-** activates Scene **A**, but deactivates Scenes **B** and **C** (this is accomplished in Control4 by activating their Toggle Scene). The Scenes may be listed in any order.

**IMPORTANT:** Activating a Selected Scene may cause the automatic deactivation of another Selected Scene. If this is the case, still include the deactivation in your Action. For example, if activating Scene **B** causes the deactivation of Scene **A**, then code = **B+A-**, even if Scene **A** will be deactivated automatically. This will ensure the LED colors are handled properly.

The driver also provides an **ELSE** concept (**Full ONLY**) where actions may be executed if the conditions are NOT met. This is normally coded with the **|** operator (the same used for 'or' in conditions). With this operator, the **ELSE** actions are executed when the conditions are NOT met, except for the time condition. This is what you would normally want: a specific program is to run during its specific time interval and the **ELSE** clause would apply only to the program's other conditions. Example: **d P+ = A | B ; n P+ C | D** where during daytime, if the security partition is armed, triggering the driver will activate Scene **A**, otherwise Scene **B**. During nighttime, Scene **C** would be activated instead when the security partition is armed, Scene **D** otherwise.

To delimit the **ELSE** clause, you may also use the **&** operator (the same used for 'and' in conditions) instead of **|**. This forces the inclusion of the time element in conditions for **ELSE** but could cause headaches with other programs using time elements. Use with care.

**SUMMARY:** Putting it all together, here are a few examples (assume **M** is a Light, **N** is a Blind, **P** is a Security Partition and room **R** has been selected). In all cases, we trigger the driver by pressing a keypad button, but other triggers may be used with the same results.

- **n M- = A** uses one program to activate Scene **A** at night, but only if light **M** (possibly in the next room) is OFF. Nothing happens during daytime.
- **(\*NEW\*) R+ = A ; R- = B** uses one program to activate Scene **A** if the selected room is currently ON (video or audio) and Scene **B** if the room is currently OFF.
- **d P+ = A ; n P+ = B ; P- W-[Sa, Su] = C ; P- W+[Sa, Su] = D** uses four programs to mean that if the partition (**P**) is armed, Scene **A** would get activated by the keypad button during the day (program 1) and Scene **B** at night (program 2). If the partition is not armed, at any time of the day, then Scene **C** would be activated except on weekends (program 3) where Scene **D** would be activated instead (program 4). It is important to make sure that your programs cover all the possibilities of what you wish to achieve.

- **$N++ = B-A ; N+ = A-B ; N- = A-B$**  uses three programs to activate Scene **A** (and not Scene **B**) if the Blind (**N**) is fully open, but Scene **B** if the blind is partially or fully closed, ensuring that Scene **A** is not active at that time. Scenes **A** and **B** might, for example, control the same lights but at different levels. The scenes are Activated / Deactivated in the order they are specified. Make sure you avoid conflicts. Alternatively, you could use the **ELSE** clause and only one program to accomplish this:  **$N++ = B-A \mid A-B$** .
- Let's say you use variable **Y** to indicate seasons, you could do the following:  
  
 **$Y [SUMMER] = A ; Y [SPRING, FALL] = B, Y [WINTER] = C$**  uses three programs to activate different Scenes (one for the Summer, a second one for Spring and Fall and a third for Winter). You simply need to set values into variable **Y** via programming, possibly triggered by the user with Experience Buttons (one for each season).
- **$Z-- \mid N- = B$**  when it is really DARK (Ambient Light) or the blind is DOWN/CLOSED, activate Scene **B**, otherwise do nothing (there are no other programs). On the keypad, the user would receive a White flash of the LED if no scene is Activated by the action (in this case, if it is not DARK or if the blind is not fully DOWN/CLOSED).
- It is possible to do a round robin of several scenes by repeatedly pressing the same keypad button (must be the 'ON' button). Using three scenes, for example:

*Control Logic When Activating/ON:*  **$A+ = C-A-B+ ; B+ = A-B-C+ ; C+ = B-C-A+ ; A-B-C- = A+$**

Scenes may be used as Conditions and the fourth program is used to initiate the first scene in the sequence if all scenes are currently Inactive. To turn everything OFF, you may have one of the scenes in the sequence do this or possibly use the Alternate feature to turn everything OFF (always with the same keypad ON button):

*Control Logic When Alternate Activating/ON:*  **$A-B-C-$**

---

## PROPERTIES

- **Cloud Status** displays the status of the DriverCentral Cloud driver.
- **Automatic Updates** turns the DriverCentral automatic update facility On or Off.
- **Driver Version** displays the version of this driver.
- **Driver Information** displays various status messages about the driver.
- **Debug Mode** turns Debug Mode Off or On (with output to the Lua Output window).
- **Debug Duration in Minutes** sets the duration of Debug On.

### Main Properties (some may not be visible):

- **Control Logic When Activating/ON:** specifies the Control Logic to be used when the user/system is activating lighting scenes (generally meaning turning lights ON). This is usually done by pressing the

ON (or Toggle) button on the connected keypad, or by turning the Virtual Light ON (by one of the usual means), or by activating the Triggering Contact, or via programming. See the **CONTROL LOGIC SPECIFICATION** below for syntax. Up to twenty different programs may be specified. If the statement you wish to use is too long for one line, you may simply end it with an ellipsis (...) and a second property will be provided to continue the statement.

- **Control Logic When Alternate Activating/ON (Medium and Full Only):** specifies the Control Logic to be used when the user/system is Activating or turning ON this special state of the driver, by either pressing a connected Alternate ON keypad button or by double tapping the connected ON (or Toggle) button or via programming. This is not available via the Virtual Light or the Triggering Contact. See the **CONTROL LOGIC SPECIFICATION** below for syntax. Up to twenty different programs may be specified. If the statement you wish to use is too long for one line, you may simply end it with an ellipsis (...) and a second property will be provided to continue the statement. **IMPORTANT:** if you do not wish to use the Alternate feature, leave this property empty (blank). Otherwise, when not using the Alternate ON keypad button, a small delay may be introduced after the ON or Toggle keypad button is pressed while the driver waits to see if the user is double tapping.
- **Behavior When Deactivating/OFF (Medium and Full Only):** specifies how the Specified Scenes will be handled when the user is deactivating lighting scenes (generally meaning turning lights OFF). This is usually done by pressing the OFF (or Toggle) button on the connected keypad, or by turning the Virtual Light OFF (by one of the usual means), or by deactivating the Triggering Contact, or via programming.
- **Control Logic When Deactivating/OFF (Medium and Full Only):** (when the Behavior property above specifies 'detailed Control Logic') specifies the Control Logic to be used when the user/system is deactivating lighting scenes (generally meaning Turning Lights OFF). See the **CONTROL LOGIC SPECIFICATION** below for syntax. Up to twenty different programs may be specified. If the statement you wish to use is too long for one line, you may simply end it with an ellipsis (...) and a second property will be provided to continue the statement.
- **When a Selected Scene is Triggered Externally:** specifies how the driver will process an unsolicited activation/deactivation of one of the Selected Scenes. This may be caused by any number of means outside of this driver, such as another keypad, the Control4 apps, the scheduler, etc.
- **Triggering Contact K Is:** specifies the normal (deactivated) state of the Triggering Contact. This contact is considered activated when it is switched to its other state and deactivated when it is switched back to its normal state.
- **Keypad LED Active/ON Color:** specifies the keypad LED color to be used when Scenes have been activated (and the Virtual Light in ON). Many popular (and generally reliable) colors are provided by name, but you may further customize the color. Be aware that the colors shown in Composer may not exactly match what will be displayed by the actual keypad. Experimentation may be necessary. Alternatively, you may also use the Top Active Color defined in the first scene activated this trigger cycle.
- **Keypad LED Inactive/OFF Color:** specifies the keypad LED color to be used when Scenes have been deactivated (and the Virtual Light in OFF). Many popular (and generally reliable) colors are provided by name, but you may further customize the color. Be aware that the colors shown in

Composer may not exactly match what will be displayed by the actual keypad. Experimentation may be necessary.

#### **Advanced Lighting Scenes and Output Contacts:**

- **Number of Advanced Lighting Scenes:** specifies how many Selected Scenes you wish to use (up to ten). The appropriate properties will then be displayed. The Selected Scenes are referred to by the letters **A** to **J**.
- **Consider only Scenes Having a Toggle Scene (Medium and Full Only):** many Advanced Lighting Scenes have a designated Toggle Scene used when you Deactivate the Scene. However, based on the Project Scene structure, you may also need to see/use Scenes without a designated Toggle Scene.
- **Scene Output Contacts Are:** specifies the normal (deactivated) state of the Output Contacts. These contacts are activated (switched to their other state) when the corresponding Scene is active and switched back to their normal state when the corresponding Scene is inactive. You may also specify this by individual contact.
- **Select Lighting Scene x (A to J):** Selects the Advanced Lighting Scene to be associated with the appropriate letter.
- **Scene x (A to J) Output Contact Is:** specifies the normal state of individual Output Contacts. The Output Contacts are activated (switched to their other state) when the corresponding Scene is active and switched back to their normal state when the corresponding Scene is inactive.

#### **Conditional Lights / Blinds / Partitions / Room / Contacts / Ambient (Full ONLY):**

- **Number of Conditional Lights/Blinds/Partitions:** specifies how many Conditional devices you wish to use (up to five). The appropriate properties will then be displayed. The Conditional devices are referred to by letters **M** to **Q**.
- **Select Light/Blind/Partition x (M to Q):** specifies which device in the Project will be monitored as this Conditional device, to be used in the **CONDITION** part of the Control Logic. Such devices may be a Light, a Blind or a Security Partition.
- **(\*NEW\*) Select Room R:** specifies which room in the Project will be monitored as this Conditional device, to be used in the **CONDITION** part of the Control Logic.
- **Conditional Contacts S to V Are:** specifies the normal (deactivated) state of the Conditional Contacts. These contacts are monitored and may be used in the **CONDITION** part of the Control Logic. They are considered active when switched to their other state and inactive when switched back to their normal state. You may also specify this by individual contact.
- **Conditional Contact x (S to V) Is:** specifies the normal (deactivated) state of each Conditional contact individually. The Conditional contacts are monitored and may be used in the **CONDITION** part of the Control Logic. They are considered active when switched to their other state and inactive when switched back to their normal state. You may also specify this by individual contact.

- **Select Ambient Light Conditional as Z:** selects a Domosapiens Ambient Light Monitor driver (available separately from DriverCentral) which may then be used in the **CONDITION** part of the Control Logic.

#### Informational Properties:

- **Selected Scene x (A to J) Details:** This informational property shows which Scene is currently associated with the corresponding letter, as well as its current state and whether it has a Toggle Scene.
- **Last Refreshed State of Conditional Lights/Blinds/Partitions (Full ONLY):** this informational property shows the most recently obtained states of Conditional devices. Such states are only refreshed when the driver is triggered to select Advanced Lighting Scenes or when the Action *Refresh Conditional States* is executed (this is useful during installation and setup).
- **Current State of Conditional Contacts (Full ONLY):** this informational property shows the current state of any connected Conditional Contacts (**S to V**). Such contacts are monitored in real time.
- **Current Value of VARIABLE\_NUM\_X (Full ONLY):** this informational property shows the current value of the driver's Conditional variable **X**. You may set the numeric value of this variable via programming and test it/use it in the Control Logic. The value may be negative and contain a decimal part.
- **(\*NEW\*) Current Value of VARIABLE\_NUM\_x (Full ONLY):** (lowercase 'x') this informational property shows the current value of the driver's Conditional variable **x**. You may set the numeric value of this variable via programming and test it/use it in the Control Logic. The value may be negative and contain a decimal part.
- **Current Value of VARIABLE\_STR\_Y (Full ONLY):** this informational property shows the current value of the driver's Conditional variable **Y**. You may set the string value of this variable via programming and test it/use it in the Control Logic.
- **(\*NEW\*) Current Value of VARIABLE\_BOOL\_y (Full ONLY):** (lowercase 'y') this informational property shows the current value of the driver's Conditional variable **y**. You may set the value of this variable via programming (True or False) and test it/use it in the Control Logic.

---

## PROGRAMMING

**(\*NEW\*)** The driver provides a programming command to trigger the selection of Scene(s), similar to pressing a keypad button. This trigger is referred to as 'p' and may be selected (or ignored) in the Control Logic. Several events are also provided, including an event where the driver needs to make sure the variables you use in the Control Logic are up to date. You may choose to update the variables in real time if this is easier, or update them when requested by the 'When Variables Requested' event.

---

## SUPPORT

For support on this driver please go to <https://help.drivercentral.io/>. Give a detailed description of the problem; also include the version number of the driver and the version of Control4 OS that you are using.

---

## AUTO UPDATE

This driver is updated with fixes and new features from time to time. To ensure your project uses the latest version, set the Automatic Updates property of the driver to On.

---

## CHANGELOG

10.0.0	May 26, 2023	Initial Release
10.0.1	May 28, 2023	Added Quick Start to documentation
11.0.0	August 7, 2023	Added Capabilities, Room, VARIABLE_BOOL_y, a second NUM variable x, negative and decimal capability for variables X and x, shorter property names
11.0.1	September 18, 2023	Fixed issues with Single/Double Tap and Scene ramping (on Single Tap)
11.0.2	January 6, 2024	Issue fixed with using Scene LED colors
11.1.0	February 13, 2024	Support for external operation of our Scenes, issue fixed with OFF Control Logic, added Alternate ON button link
12.0.0	January 19, 2025	Added scene ramping, minor issues fixed
12.1.0	February 2, 2025	Added support for scenes which do not support ramping, minor issues fixed