

# VIRTUAL OBJECT – CALENDAR MANUAL

## Object Manager Configuration

### TABLE OF CONTENT

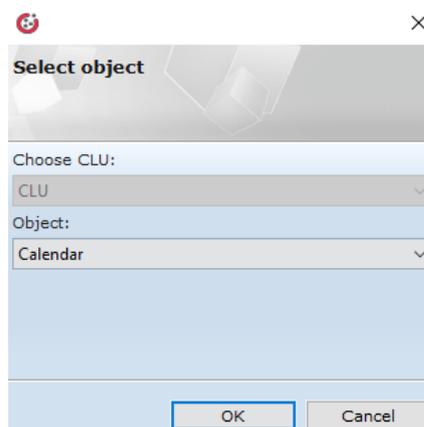
1. ADDING VIRTUAL OBJECT TO THE CONFIGURATION	1
2. VIRTUAL OBJECT CONFIGURATION	2

## 1. ADDING VIRTUAL OBJECT TO THE CONFIGURATION

1. Open Object Manager configuration software
2. Create new project and execute *CLU Discovery*/ Open an existing project and communicate with CLU
3. From the main menu select option Add *CLU Virtual Object*



4. Select object and confirm by clicking OK button



5. Enter any name of object and confirm by clicking OK button

**ATTENTION!** Names of virtual objects cannot contain spaces.

6. Having created an object, a configuration window of virtual object will appear on the screen Calendar - it has 4 tabs
7. The window may be closed and the configuration may be sent at the stage of created virtual object not steered by anything or you may proceed to step Configuration of Virtual Object

## 2. VIRTUAL OBJECT CONFIGURATION

### 1. Basic Configuration

- a. The virtual object Calendar enables conduct of the set actions at specified times of day, week, month or year, depending on the agreed rule
- b. Go to *Rule* tab in order to establish moments in time you are interested in, when you wish to carry out a specific action (see: point 2)
- c. Go to Events tab. To the *OnCalendar* event assign the method or launching the script you are interested in to a specific calendar rule
- d. Send configuration to CLU (in order to complete more detailed configuration you may use the following tips)
- e. Having configuration sent, make sure that the schedule is in operational mode (feature *State* is set on value 1-see point 7)

### 2. Rule Tab

- a. This tab enables setting up moments in time when you wish to steer a device (i.e. Schedule garden watering)-by setting up a rule we organize the time when an event will be invoked *OnCalendar*
- b. Settings carried out in Rule tab are analysed in line with the standard Cron - to find out how it works, we recommend the use of internet schedules to check the correctness of settings<sup>1</sup> - for the site indicated in the link you may paste the rule set up in the program in order to check its correctness

---

<sup>1</sup> An example may be the following website <http://crontab.guru>

- c. Placing the rules in the tab is divided into two groups-those specifying day and time. Configuration of each of the groups has an impact on automatic set up of the feature Rule. Settings from each of the groups are combined with one another according to the rule of adding new conditions which do not impact the previously entered ones-that means entering a condition concerning for example 21 day of the month will not collide with the settings of even days
- d. Fields visible on the above graphics Month, Day of week and Even/odd days post activation 0 enable setting according to our will to set up an event *OnCalendar*
- e. Field *Day of month* after activation  enables setting of days of month in which we want to evoke an event *OnCalendar*. Days we are interested in are entered successively, after comma, or by indicating a range, i.e. 6-20 Both methods of introduction may be combined, i.e. 5,7,10,12-16
- f. Field *Time* is split separately into hours and minutes. Both fields assume values as:
  - Values entered individually, after the comma or as a scope (i.e. 6; 5,6,7,8; 4 - 20) marking individual hours. Both methods of entering may be combined, i.e. 5-8,10,12
  - Indicating the range of hours or minutes between which we wish to carry out an event by means of a tag „\*/" (i.e. \* indicates every hour/minute", \*/3 indicates „every 3 hours/minutes)
- ATTENTION!** Settings made in Rule tab operate in such a way that after selecting conditions of all settings of time and days, their execution is combined, which means that the selection of even days, month of February and Mondays, will cause an event on all Mondays and even days of the month of February.
- g. Unselecting one of the settings (unchecking), causes ignoring of this particular setting, which means that lack of the set month and unchecking of the selection upon the field Month will be treated as a choice of all months-lack of additional conditions
- h. Sample setting I:

That setting is represented in *Rule* Feature as following:

**15   \*/2   2-18,2-30/2   2,4,7,9   5,6**  
**A   B   C   D   E**

Gdzie:

- A – chosen minutes – only 15 minutes after chosen hours
- B – chosen hours – each two hours counted from 00:00 in chosen days
- C – chosen days – single field responsible for entering from field Days and specification of parity. Record applied here 2-18,2-30/2 means the scope of days from 2 to 18. Element 2-30/2 means even days. Record 2-16/3 would mean the values 2, 5, 8, 11, 14 - scope and interval are entered which occur in between the subsequent days
- D – chosen months – counted from 1 to 12
- E – chosen days of week – counted from 1 to 7

Wybrane Selected setting means that an event will be made:

*15 minutes After each second hour*

*Each day of the month from 2. to 18. and each other even day and on Fridnays and Saturdays*

*In February, April, July and September*

### 3. Embedded Features Tab

- a. The tab enables a preview of values specifying a virtual object
- b. Feature Rule is a sequence of signs which defines a rule of calendar introduced in the Rule tab- it may be copied and pasted to the method *SetRule* so that after making changes it was possible to return easily and fast to the former settings- it is generated in accordance with the standard Cron, therefore it may be entered manually. Upon incorrect setting of a feature *Rule* with the use of invoking the method *SetRule*, feature displays ERROR
- c. Feature Rule is initially filled out as „ \* \* \* \* \* “, where each star sign signifies specific field set in the Rule tab. If a setting is missing, „ \* “ should be placed in its place.
- d. Feature *SinceLastRun* presents time which passed from the occurrence of the last event in minutes *OnCalendar* according to the set rule
- e. Feature *ToNextRun* presents time which is left until occurrence of the next event in minutes *OnCalendar* according to the set rule
- f. Feature *State* defines the state of virtual object. State of value 1 signifies that the object analyses, works and transfers the set rule to control the device steered by the calendar (see: point 3b) Value 0 indicates that even if a calendar rule is set (feature *Rule* and Rule tab *OnHarmonogram* is not evoked

### 4. Events Tab

- a. Virtual Object reacts to 4 events, to which one may assign  specific actions which may be carried out after occurrences of events
- b. Event *OnCalendar* is evoked in accordance with the set rule in the Rule tab, which means that when setting the calendar specific moments in time are selected in which we wish to carry out a given action (such as activate watering the garden)-the method of setting the rule is specified in point 1.
- c. Event *OnStart* is triggered at the time of changing of the feature *State* from 0 to 1 - launching of schedule actions
- d. Event *OnStop* is triggered at the time of changing the feature *State* from 1 to 0 - ceasing the schedule operations
- e. Event *OnCancel* triggered at the time when the upcoming event set as a result is cancelled (if we use the method *CancelNext* - see: point 4e)

### 5. Methods Tab

- a. This tab contains 4 methods which may be set as actions for occurrence of specific events, or they may be evoked by clicking on the button  by a given method
- b. Method *Start* is designated for launching the operations of virtual object. Evoking the method for setting feature *State* of the object on the value of 1
- c. Method *Stop* is designated to cease operations of the virtual object. Evoking it sets the value of feature *State* to 0
- d. Method *SetRule* is designated at setting up the feature *Rule* using a sequence of symbols in accordance with the standard Cron (method of setting up the rule is described in point 1.). At

the time of entering incorrect sequence of signs and evoking the method, within the feature *Rule* one must enter *ERROR*

**ATTENTION!** If after evoking the method, CLU is reset, the feature *Rule* will be set as at the initial completion and submission of configuration at CLU

- e. Method *CancelNext* is designated for cancelling the upcoming event set for an event *OnCalendar* (i.e. cancelling the closest garden watering due to change in weather). Upon its evoking the number of closest events which we wish to cancel should be indicated

6. After finishing the configuration, remember about sending it to the CLU



- 7. Make sure that the features State for all created calendars are equal 1. To check, double click on the virtual object Calendar on the list of modules and proceed to tab Features. Otherwise, launch it by means of method Start in each calendar (see: point 5b)