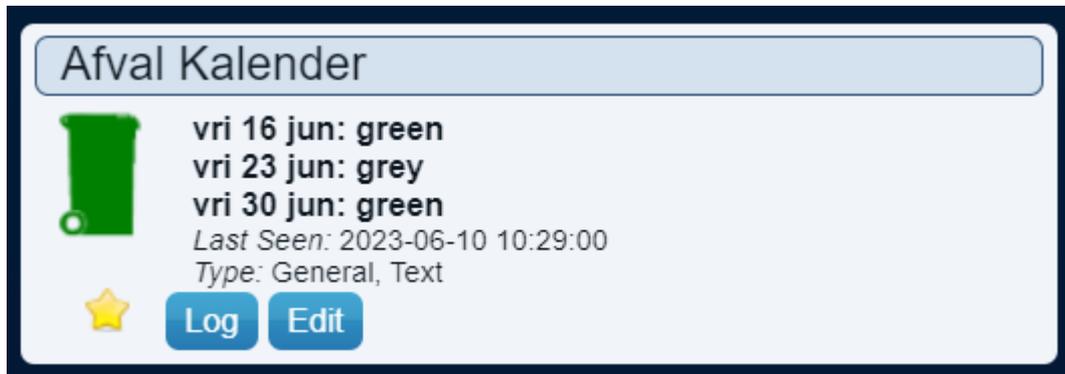


GarbageCalendar



This is a project to retrieve the Garbage calendar information for your home address by means of a Domoticz time script which will update a Domoticz TEXT device and optionally send you a notification at the specified time(s) 0-x days before the event. This repository is a replacement for the initial repository I started: <https://github.com/jvanderzande/mijnafvalwijzer>. The main changes are:

- This repository is modular making it easier to maintain and add new website modules for other municipalities.
- It has a single main script called "`script_time_garbagecalendar.lua`"
- It will run both as Time Event LUA script or as DzVents script.
- Subdirectory "`garbagecalendar`" contains all available modules for the supported municipality websites and your personal configuration file "`garbagecalendarconfig.lua`".
- The selected module scripts is ran one time per day in the background to get the website data and save that to a Datafile.
- This Data file is used by the mainscript at the requested times.
- Add your own LUA/DzVents logic for Notifications to e.g. switch on a RGB bulb in a particular color.
- No hanging Domoticz event system, generating a "longer than 10 seconds" error when the website is unresponsive, as the Webupdate is running in the background in its own process.
- The script has much more error checking and standard logging to make problem solving much simpler.
- The main script can optionally create an ics calendar file which can be used by a calendar application.
- It now also has the option to show an custom icon for the text device for the first upcoming garbage type.

Check the Wiki for:

- [How does it all work](#)
- [Modules](#)
- [Setup instructions](#)
- [Add own notification code](#)
- [Test and Debugging instructions](#)

More information can be found or questions can asked here:

[Domoticz forum thread:](#)

[Github discussions section:](#)

PullRequest for Fixes/Changes/Additions must be made against the development branch so it can be tested first. Thanks :-)

GarbageCalendar is a free and open source application written in lua and distributed under the [GNU General Public License](#).

How does it work?

The main script "`script_time_garbagecalendar.lua`" needs to be stored in the "`domoticz/scripts/lua`" or "`domoticz/scripts/dzvents/scripts`" directory which will cause it to be triggered each minute.

The main script will load your personal configuration from "`garbagecalendar/garbagecalendarconfig.lua`", where you have defined all required information as explained in the [Setup Wiki page](#).

It will read the information from the variable "`garbagetype_cfg`" table to check for the specified update/notification times defined, and when one of these times is equal to the current time, the following things will happen:

1. When the `garbagetype` field contains "reloaddata", a background batchjob will be submitted to update the data without holding up the event system. The retrieved data will be stored in file `***"garbagecalendar_modulename.data"***` and the retrieval process information in `***"garbagecalendar_modulename_run_webupdate_backgound.log"`. **They will both be located in the directory you specified for `***"datafilepath"`.** This process will be ran in the foreground immediately in case the `***"garbagecalendar_modulename.data"***` doesn't exist yet and one of the check times is valid, to ensure the data is there to work with.
2. The next number of events, as defined with `ShowNextEvents`, will be merged into a text field, with the format as defined with `textformat`.
3. Update the defined Domoticz Text Dummy Device with this formatted text.
4. When it contains a valid Garbage type and the `hour` & `min` are equal to the current time, a notification will be send as specified in the `garbagecalendarconfig.lua` file.

Available Garbage calendar collection modules

m_burgerportaal

This script is for the site 21burgerportaal.mendixcloud.com used by: assen, bar, rmn. Set the config field `PBPName` to one of these: 'assen'/'bar'/'rmn'.

m_deafvalapp

This script is for the site deafvalapp.nl for the municipalities Bergeijk, Bladel, Boekel, Boxmeer, Buren, Cuijk, Culemborg, Echt-Susteren, Eersel, Geldermalsen, Grave, Helmond, Lingewaal, Maasdiel, Mill en Sint Hubert, Neder-Betuwe, Neerijnen, Oirschot, Reusel-De Mierden, Sint Anthonis, Someren, Son en Breugel, Terneuzen, Tiel, West Maas en Waal, Zaltbommel.

m_mijnafvalwijzer & m_mijnafvalwijzer_api

This script is for the site mijnafvalwijzer.nl and originally based on the idea from this thread/posts by [zicht @ Sharing : Afval kalender in text sensor - Domoticz](#) and script version by [nf999 @ Sharing : Afval kalender in text sensor - Page 3 - Domoticz](#). This script is also used for **Rova**, which requires this in the config: `Hostname = 'afvalkalender.rova.nl'`. Used for the municipalities supported by Rova like: Amersfoort, Bovenveld, Raalte, Steenwijk, Winterswijk and Zwolle.

m_montferland

This script is for the site www.montferland.afvalwijzer.net used by the municipality Montferland.

m_omrin

This script is for the site [Afvalkalender | Omrin](http://Afvalkalender.Omrin) used by municipalities in Friesland.

m_opzet & m_opzet_api

The `m_opzet.lua` module script is a generic screen scraping script for all municipalities that use the service of OPZET.NL. check their website for supported municipalities: http://www.opzet.nl/afvalkalender_digitaal or check the afvalkalender of your municipality and check whether it refers at the bottom of the page to "Ontwerp & techniek: Opzet". The `m_opzet_api.lua` script is created for those municipalities that use OPZET but do not support the screen scraping (`m_opzet.lua`) version on their website.

m_recycleapp-be

This script is for the site recycleapp.be (use to be www.ophaalkalender.be) for the municipalities supported by <https://www.fostplus.be>

m_rova_api (decommisioned)

Please use `m_mijnafvalwijzer` & `m_mijnafvalwijzer_api` and changewhich requires this in the config: `Hostname = 'afvalkalender.rova.nl'`

Used for the municipalities supported by Rova like: Amersfoort, Bovenveld, Raalte, Steenwijk, Winterswijk and Zwolle.

m_ximmio

This script is for the site <https://wasteapi.2go-mobile.com/api> for the municipalities supported by [Referenties - XIMMIO](#) like Almere.

m_zuidlimburg (RD4)

This script is for the site www.rd4info.nl for the municipalities in Zuid Limburg.

Setup

](https://github.com/jvanderzande/GarbageCalendar/wiki/Setup/_history)

Setup instructions

Prerequisites

- `lua` must be installed which can be checked by running the following command: `lua -v`
Install lua52 in case it isn't installed yet.
- `curl` must be installed which can be checked by running the following command: `curl --version`

Setup steps

- **Domoticz**
- Create a Dummy Text Sensor in Domoticz and give it an appropriate name: eg "Container".
This Text device will be filled with the upcoming garbage collection schedule.
- **Install scripts**
- Download [the project zipfile](#) and unzip the main script "`script_time_garbagecalendar.lua`" and "`garbagecalendar`" directory one of these 2 directories:
 - "`domoticz/scripts/lua`" directory to use the standard LUA event system.
 - or-
 - "`domoticz/scripts/dzvents/scripts`" directory in case you want to run the script with **DzVents**

In case you are using the Domoticz Docker container, you simply copy the scripts into container directory:

- "`/opt/domoticz/config/scripts/lua`" directory to use the standard LUA event system.
- or-
- "`/opt/domoticz/scripts/dzvents/scripts`" directory in case you want to run the script with **DzVents**

The data directory will be created in the garbagecalendar subdirectory by default, which will contain the logs and retrieved data.

You could also specify your own data directory in the config file as defined in a later step on this page.

- **Create personal config**

- Go into directory "garbagecalendar/" and copy file "garbagecalendarconfig_model.lua" to "garbagecalendarconfig.lua" to create your personal config file. "garbagecalendarconfig.lua" will contain your personal setup and preferences used by the process. This file isn't updated when you upgrade the scripts from Github, so you will have to do this manually in case of changes/additions.

- Update your personal configuration

Edit the following information in file "garbagecalendarconfig.lua":

- Domoticz & Address information

```

-- Specify your information here as needed for your needs
myGarbageDevice = '' -- The Text devicename in Domoticz. eg 'Container'
Zipcode = '' -- Your zipcode eg '1234AB'
Housenr = '' -- Your housnr. eg '99'
Housenrsuf = '' -- Your housnr suffix ('a', 'b' etc)
Hostname = '' -- m_opzet & m_opzet_api: Specify the hostname of your website. eg
'inzamelkalender.hvcgroep.nl'
Street = '' -- Street name Only needed for: m_recycleapp
BPName = '' -- Burgerportaal name either: 'assen'/'bar'/'rmn'
Companycode = '' -- m_ximmio Companycode. (See m_ximmio.lua file for how to find this code).

```

Required information per module: (R=Required/O=Optional)

Module	Zipcode	Housenr	Housenrsuf	Hostname	Street	BPName	Companycode**
m_burgerportaal	R	R	O			R	
m_deafvalapp	R	R	O				
m_mijnafvalwijzer	R	R	O	O			
m_Montferland	R	R	O				
m_omrin	R	R	O				
m_opzet	R	R	O	R			
m_opzet_api	R	R	O	R			
m_recycleapp-be	R	R	O		R		
m_ximmio	R	R	O				R
m_zuidlimburg	R	R	O				

- Modules

choose one of these modules by removing the "--" in front of the modulename. eg for "mijnafvalwijzer":

```

--websitemodule = "m_deafvalapp"
websitemodule = "m_mijnafvalwijzer"
--websitemodule = "m_mijnafvalwijzer_api"
--websitemodule = "m_montferland"
--websitemodule = "m_recycleapp"
--websitemodule = "m_omrin"
--websitemodule = "m_opzet"
--websitemodule = "m_opzet_api"
--websitemodule = "m_recycleapp-be"
--websitemodule = "m_ximmio"
--websitemodule = "m_zuidlimburg"
----Special module read heading in the file for details
--websitemodule = "m_csv_file"

```

directories to used

Specify the appropriate directories (Raspberry Pi example):

```
-- Specify the directory used for the Data and Log files.
-- It will default to the gabagecalendar/data directory when invalid or missing.
--datafilepath      = '/var/tmp'
```

- In case you run Domoticz in Docker you could simply leave datafilepath empty!
This will trigger the Data and logfiles to be stored in: `/opt/domoticz/config/scripts/lu/gabagecalendar/data`
- Notifications, device text and device icon

To allow for notifications and alternative text in the domoticz device, you need to define a line for each garbage type returned by the webrequest. To determine what these garbagetypes should be, you simply run the script the first time with variable `"mydebug = true -- (true/false)"`. This will always process the datafile and tell you about the missing garbage types in the domoticz log like this:

```
13:45:00 MainScript: #!# -- start -- Add these records into the garbagetype_cfg table and adapt the schedule and
text info to your needs :
  ["green"]      = {hour=19,min=02,daysbefore=1,reminder=0,text="green",icon=""}, ["paper"]      =
{hour=19,min=02,daysbefore=1,reminder=0,text="paper",icon=""}, ["grey"]      =
{hour=19,min=02,daysbefore=1,reminder=0,text="grey",icon=""},
13:45:00 MainScript: #!# -- end -----
```

So you simply copy and paste those (3) lines from the log into the below section of `"garbagecalendarconfig.lua"` and update the text field to the description you like to see in the Domoticz text device and the hour & min fields to the time you like to get a notification:

```
-- Look at the Domoticz log for any missing records as they will be displayed there and can be just copy/pasted in.
-- Fields description:
-- hour & min ==> the time the check needs to be performed and notification send when daysbefore is true
-- daysbefore ==> 0 means that the notification is send on the day of the planned garbage collection
-- daysbefore ==> X means that the notification is send X day(s) before the day of the planned garbage
collection
-- reminder ==> Will send a second reminder after x hours. 0=no reminder (can be between 0 - 24 hours)
-- text ==> define the text for the notification and Text Device.
-- icon ==> (optional) define the icon to use for the text device in Domoticz
-- active ==> (optional) default="on"
-- active="on" Both will show in Domoticz Text Device and Send notifications
-- active="off" Will show in Domoticz Text Device but not send notifications
-- active="skip" Will not show in Domoticz Text Device and not send notifications

-- The "reloaddata" entry is required to run the background process to update the data one time per day.
-- The "dummy" entry can be used to force reading the data, update the Domoticz text device and see if there are
any errors or missing garbage types.

garbagetype_cfg = {
-- Add any missing records below this line
["green"]      = {hour=19,min=02,daysbefore=1,reminder=3,text="Groene bak",icon="garbagecalendar_green"},
["paper"]      = {hour=19,min=02,daysbefore=1,reminder=3,text="Papier",icon="garbagecalendar_blue"},["grey"]
={hour=19,min=02,daysbefore=1,reminder=3,text="Restafval",icon="garbagecalendar_grey"},
-- Add any missing records above this line
-- "reloaddata" is used to start the background update process at this given time.
["reloaddata"] = {hour=02,min=30,daysbefore=0,reminder=0,text="trigger for reloading data from website into
garbagecalendar_modulesname.data"},["dummy1"]      = {hour=02,min=30,daysbefore=0,reminder=0,text="dummy to
trigger testing"}}
```

The icon="" field is either empty or needs to contain the name of one of the already existing Domoticz Custom Icons or one of the available icon.zip files in the icons directory. You can add your own preferred icons there too and use that name in icon= config field. Garbage_calendar will look in the `"garbagecalendar/icons"` directory for any missing icons and try to upload it into

Domoticz when available. The text device icon will be set to the default (0) when there is no icon defined or the defined Icon isn't available in Domoticz.

Update the Configuration section for the Notification system in case you like to get warned at the defined times in "garbagetype_cfg":

```
NotificationEmailAddress = {'',''} -- Specify multiple Email Addresses for the notifications. Leave empty to skip
email notification
Notificationssystem = '' -- Specify notification system eg
"telegram/pushover/gcm/http/kodi/lms/nma/prowl/pushalot/pushbullet/pushsafer" leave empty to skip
Notificationscript = '' -- Specify personal notification script/command eg: lua sendmessage.lua "@TEXT@"
(where @TEXT@ will be replaced by the notification text.)
EventNotificationscript = '' -- Script to run in the Domoticz event system when a notificcate occurs. See example
"notification_event_script.lua"
--
It also supports: @GARBAGETYPE@; @GARBAGEDATE@; @GARBAGETEXT@
-- Define how the title and bodytext should look
-- @DAY@ ==> Will be replaced by notificationtoday; notificationtomorrow; notificationlonger depending on
the days difference.
-- @GARBAGETYPE@ ==> Will be replaced by the GarbageType definition from the WebSite
-- @GARBAGEDATE@ ==> Will be replaced by the pickup date found on the schedule data
-- @GARBAGETEXT@ ==> Will be replaced by the text from garbagetype_cfg[].text field
-- @REMINDER@ ==> Will be set to "" for first notification and "notificationreminder" for the reminder run
-- @TEXT@ ==> Will be replaced by the finalised notificationtext and can be used in the notification script
-- @TITLE@ ==> Will be replaced by the finalised notificationtitle and can be used in the notification script

--### Dutch example
notificationreminder = 'herinnering'
notificationtitle = 'GarbageCalendar: @DAY@ de @GARBAGETEXT@ aan de weg zetten!'
notificationtext = '@GARBAGETEXT@ wordt @DAY@ opgehaald! (@REMINDER@)'
notificationtoday = 'vandaag'
notificationtomorrow = 'morgen'
notificationlonger = 'over @DAYS@ dagen'
notificationdate = 'wd dd mmmm yyyy' -- @GARBAGEDATE@ format -> Options are the same as available for
textformat date options

--### English example
--~ notificationtitle = 'GarbageCalendar: @GARBAGETEXT@ will be picked up in @DAY@!'
--~ notificationtext = 'Put the @GARBAGETEXT@ out as it will be picked up @DAY@.!'
--~ notificationtoday = 'today'
--~ notificationtomorrow = 'tomorrow'
--~ notificationlonger = 'in @DAYS@ days'
--~ notificationdate = 'wd dd mmmm yyyy' -- Options are the same as available for textformat date options
```

Format text for Domoticz device

Optionally updated the default text format for the Domoticz text device in this section::

```
-- ### define format for text device
-- date options:
-- wd  = weekday in 3 characters as defined in the daysoftheweek table below. eg Zon;Maa;Din
-- wdd = weekday as defined in the Longdaysoftheweek table below. eg zondag;maandag;dinsdag
-- dd  = day in 2 digits   eg 31
-- mm  = month in 2 digits eg 01
-- mmm = month abbreviation in 3 characters as defined in the ShortMonth table below. eg : jan
-- mmmm = month as defined in the LongMonth table below. eg: januari
-- yy  = year in 2 digits eg 19
-- yyyy = year in 4 digits eg 2019
-- Garbage type description options
-- sdesc = short garbage type description from Website   eg pmd
-- ldesc = long garbage type description from Website when available, will be replaced by table description when
not
-- tdesc = Use the description available in the table text field
textformat = "wd dd mmm - tdesc"
```

Standard conversion tables

Optionally Define/update the weekday, short abbreviation to be shown in the text device. see previous topic:

```
-- Date/day info:
daysoftheweek={"zon","maa","din","woe","don","vri","zat"}
Longdaysoftheweek={"zondag","maandag","dinsdag","woensdag","donderdag","vrijdag","zaterdag"}
ShortMonth={"jan","feb","maa","apr","mei","jun","jul","aug","sep","okt","nov","dec"}
LongMonth=
{"januari","februari","maart","april","mei","juni","juli","augustus","september","oktober","november","december"}
```

upcoming events to shown

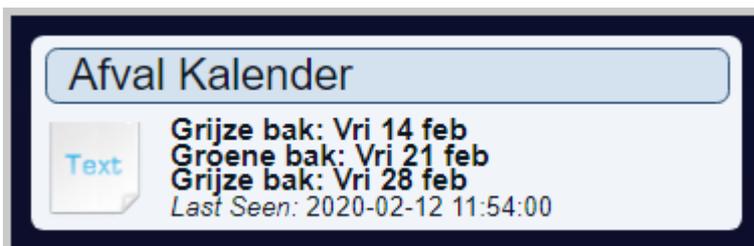
Optionally define whether you like to see the upcoming event including duplicate garbage types or the next x events of unique garbage type events:

```
-- ### define what to show in the Domoticz text device
-- false => show multiple occurrences of a garbagetype
-- true  => show one the next occurrence for a unique garbagetype
ShowSinglePerType = false
```

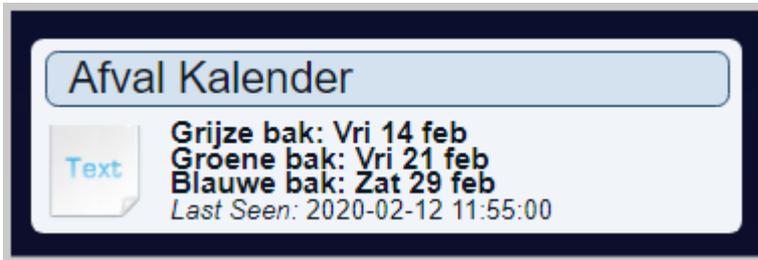
example result difference for *ShowSinglePerType* with these upcoming events:

```
Grijze bak - 14-02-2020
Groene bak - 21-02-2020
Grijze bak - 28-02-2020
Blauwe bak - 29-02-2020
```

- *ShowSinglePerType* = false



ShowSinglePerType = true



- Generate ICS file

Optionally enable the generation of an ics calendar file:

```
-- Configuration for the generation of an ics file:
-- IcalDesc:
-- @GARBAGETYPE@ ==> Will be replaced by the GarbageType definion from the WebSite
-- @GARBAGETEXT@ ==> Will be replaced by the text from garbagecalendar_cfg[].text field
IcalEnable = false           -- false/true: When true, a garbagecalendar_Modulename.ics will be created in the
datafilepath which can be used in a calendar application.
IcalTitle = "GarbageCalendar" -- title of the calendar
IcalDesc = "@GARBAGETEXT@ wordt opgehaald." -- text for the events in the calendar
IcalEvents = 10             -- max number of upcoming events to save to icalfile, but could be less when less
events are provided by the website.
IcalNotify = 12             -- Notification Time in hours before event. 0=no notification
```

We're done

You should be ready to go now. Ensure you reset the mydebug to:

-- run the script as it normally does when any of the scheduled times is the current time

mydebug = false -- run the web update module in the foreground with each run for debugging.

testdataload = false

-- run the web update module in the background with each run for debugging. (testdataload takes presedent)

testdataloadbatch = false

-- trigger a test notification each run for the first record for testing the notification system

testnotification = false

to avoid a lot of messages in the Domoticz log. check the [testing](#) page in case you need to test/debug your installation.

Notifications

config notification script

1. Use Event system script

Edit the following information in file "garbagecalendarconfig.lua":

1. Update garbagecalendarconfig.lua:

- ```
-- Specify personal notification event script located in GarbageCalendar directory
-- This example is provided and will work in DzVents and regular Lua Time Events
EventNotificationscript = 'notification_event_script.lua'
```

- Available variables that can be used in the scriptcode:

- Notify\_type = GarbageType as defined as key in "garbagetype\_cfg" and coming from the internet.  
Notify\_text = Formed text sentence using the settings from the "notificationtext" field  
Notify\_date = Formatted notification date using the config variable "notificationdate"  
Notify\_mtitle = Formatted Title sentence using the settings from the "notificationtitle" field  
Notify\_mtext = Description from "garbagetype\_cfg" "text" field  
Notify\_reminder = Formatted content for the config variable "notificationreminder"

- Edit the "notification\_event\_script.lua" script:

This script currently contains an example to switch an RGB lamp on for xx seconds and set the color depending on the garbagetype. The actions are by default disabled and the script return's always on line 15 without performing any action. To active the script just comment or remove line 15! It contains both the logic when ran in DzVents or as regular LUA Time Event.

```

function notification_event(RunbyDzVents, commandArray, domoticz) -----

-- garbagecalendar notification script
-- Example script which will Switch a light on with specified color in case of a garbagecalendar notification
--[[=== Available variables from the mainscript :
 Notify_type = GarbageType as defined as key in "garbagetype_cfg" and coming from the internet.
 Notify_text = Formed text sentence using the settings from the "notificationtext" field
 Notify_date = Formatted notification date using the config variable "notificationdate"
 Notify_mtitle = Formatted Title sentence using the settings from the "notificationtitle" field
 Notify_mtext = Description from "garbagetype_cfg" "text" field
 Notify_reminder = Formatted content for the config variable "notificationreminder"
]]

--##### To enable the script you have to remove/comment the next line #####
if 1==1 then return end

-- ensure they are all initialised to avoid errors.
Notify_type = Notify_type or ""
Notify_text = Notify_text or ""
Notify_date = Notify_date or ""
Notify_mtitle = Notify_mtitle or ""
Notify_mtext = Notify_mtext or ""
Notify_reminder = Notify_Reminder or ""

-- Start you logic here in the appropriate section

local lamp = 'TestLight'
local ontime = 30

if RunbyDzVents then
 -- #####
 -- DzVents logic part will only run when implemented in DzVents
 -- #####
 print('DzVents Type:' .. Notify_type ..
 ' text:' .. Notify_text ..
 ' Date:' .. Notify_date ..
 ' message title:' .. Notify_mtitle ..
 ' message text:' .. Notify_mtext ..
 ' Reminder:' .. Notify_reminder
)
 print('Turn ' .. lamp .. ' on with a Color on for '..ontime..' seconds') --## Turn on lamp xx
percent
 domoticz.devices(lamp).switchOn() --## setColor(r, g, b, br, cw, ww, m, t)
 if Notify_type == 'gft' then --green
 domoticz.devices(lamp).setColor(0, 255, 0, 50, 0, 0, 3, 0)
 elseif Notify_type == 'papier' then --blue
 domoticz.devices(lamp).setColor(0, 0, 255, 50, 0, 0, 3, 0)
 elseif Notify_type == 'pmd' then -- yellow
 domoticz.devices(lamp).setColor(255, 255, 0, 50, 0, 0, 3, 0)
 elseif Notify_type == 'restafval' then -
 domoticz.devices(lamp).setColor(128, 128, 128, 100, 0, 0, 3, 0)
 else -- purple
 domoticz.devices(lamp).setColor(128, 0, 128, 50, 0, 0, 3, 0)
 end

 --## Reset to WarmWhite after xx seconds
 domoticz.devices(lamp).setColor(0, 0, 0, 20, 0, 0, 2, 255) .afterSec(ontime-5) --## Turn off lamp after
xx seconds when it is currently off
 if domoticz.devices(lamp).state == "Off" then
 domoticz.devices(lamp).switchOff().afterSec(ontime) end
else
 -- #####
 -- LUA Time Event script, will only run when Lua Time Script
 -- #####
 print('LUA Time:' .. Notify_type ..
 ' text:' .. Notify_text ..
 ' Date:' .. Notify_date ..
 ' message title:' .. Notify_mtitle ..
 ' message text:' .. Notify_mtext ..
 ' Reminder:' .. Notify_reminder
)
 print('Turn ' .. lamp .. ' on with a Color on for '..ontime..' seconds') --## get idx for lamp

```

```

idx=otherdevices_idx[lamp] --## Turn on lamp
commandArray[#commandArray + 1] = {[lamp] = 'On'} --## setColor(r, g, b, br, cw, ww, m, t)
url = 'http://127.0.0.1:8080/json.htm?type=command¶m=setcolbrightnessvalue&idx='..idx
if Notify_type == 'gft' then --green
 url = url .. '&color={"m":3,"t":0,"r":0,"g":255,"b":0,"cw":0,"ww":0}&brightness=50'
elseif Notify_type == 'papier' then --blue
 url = url .. '&color={"m":3,"t":0,"r":0,"g":0,"b":255,"cw":0,"ww":0}&brightness=50'
 domoticz.devices(lamp).setColor(0, 0, 255, 50, 0, 0, 3, 0) elseif Notify_type == 'pmd' then -- yellow
 url = url .. '&color={"m":3,"t":0,"r":255,"g":255,"b":0,"cw":0,"ww":0}&brightness=50'
 domoticz.devices(lamp).setColor(255, 255, 0, 50, 0, 0, 3, 0) elseif Notify_type == 'restafval' then -
-
 url = url .. '&color={"m":3,"t":0,"r":128,"g":128,"b":128,"cw":0,"ww":0}&brightness=50'
 domoticz.devices(lamp).setColor(128, 128, 128, 100, 0, 0, 3, 0) else -- purple
 url = url .. '&color={"m":3,"t":0,"r":128,"g":0,"b":128,"cw":0,"ww":0}&brightness=50'
 domoticz.devices(lamp).setColor(128, 0, 128, 50, 0, 0, 3, 0) end
commandArray[#commandArray + 1] = [{"OpenURL"} = url] --## Reset color after xx seconds via URL
url = url .. '&color={"m":2,"t":255,"r":0,"g":0,"b":0,"cw":0,"ww":0}&brightness=20'
url = url .. ' AFTER '..ontime
commandArray[#commandArray + 1] = [{"OpenURL"} = url] --## Turn off lamp after xx seconds when it is
currently off
if otherdevices['yourotherdevicename'] == "Off" then
 commandArray[#commandArray + 1] = {[lamp] = 'Off AFTER '..ontime} end
end
end

```

## Use an async External process/script

Edit the following information in file "garbagecalendarconfig.lua":

1. Available variables that can be used in the commandline:

- ```

-- Supported variables for Title and Body
-- @DAY@      ==> Will be replaced by notificationtoday; notificationtomorrow; notificationlonger depending
on the days difference.
-- @GARBAGETYPE@ ==> Will be replaced by the GarbageType definition from the WebSite
-- @GARBAGEDATE@ ==> Will be replaced by the pickup date found on the schedule data
-- @GARBAGETEXT@ ==> Will be replaced by the text from garbage_type_cfg[].text field
-- @REMINDER@  ==> Will be set to "" for first notification and "notificationreminder" for the reminder run
-- @TEXT@      ==> Will be replaced by the finalised notificationtext and can be used in the notification
script
-- @TITLE@     ==> Will be replaced by the finalised notificationtitle and can be used in the notification
script

```

- Update garbagecalendarconfig.lua:

- ```

-- Specify personal notification script/command eg: lua sendmessage.lua "@TEXT@" (where @TEXT@ will be
replaced by the notification text.)
---Notificationscript = 'lua /home/pi/domoticz/scripts/lua/garbagecalendar/notification_external_script.lua
"@GARBAGETYPE@" > /tmp/GC_Notify.log 2> /tmp/GC_Notify.log'

```

- Edit the script:

- 1.

```

-- garbagecalendar external notification script
-- example for this setting which will switch on a CCT (RGB&CC/CW) lightbulb on a selected color at
notification time:
-- Notificationscript = 'lua notification_extenal_scrint.lua "@GARBAGETYPE@" > /tmp/GCexternal.log'

=====
-- get general functions script directory
function script_path() return arg[0]:match('.*[\\\/]') or './'
end
print(script_path() or '')
-- include generalfunctions library
dofile((script_path() or '') .. 'generalfuncs.lua') --

-- Variables
local domoticzurl = 'http://127.0.0.1:8080' -- define the url for domoticz
local idx = 439 -- define the IDX of the Light
afwlogfile = '/tmp/GC_ExEvent.data' -- datafile voor perform_webquery()
local red = 0 -- Red color 0-255
local green = 0 -- Green color 0-255
local blue = 0 -- Blue color 0-255

-- Get first commandline parameter => garbagetype
local garbagetype = arg[1] or ''
print('garbagetype:' .. (garbagetype or ''))

-- Start logic
if garbagetype == 'paper' then
 -- Turn on lamp and set color to blue
 blue = 255
elseif garbagetype == 'gft' then
 -- Turn on lamp and set color to green
 green = 255
end

-- Proces when a color was defined
if red ~= 0 or green ~= 0 or blue ~= 0 then
 print('Colors (r g b):', red, green, blue) -- Set lamp to the defined color and brightness
 -- See the Wiki for the proper definition you might require for your lightbulbs:
 --
 https://www.domoticz.com/wiki/Domoticz_API/JSON_URL's#Set_a_light_to_a_certain_color_or_color_temperature
 brightness = 100 -- (0-100%)
 url = " --data-urlencode 'color={\"m\":3,\"t\":0,\"r\":0,\"g\":0,\"b\":0,\"cw\":0,\"ww\":255}' "
 .. "}" "
 url = url .. ' ' .. domoticzurl .. '/json.htm?type=command¶m=setcolbrightnessvalue&idx=' .. idx ..
 '&brightness=' .. brightness .. ''
 print(url) perform_webquery(url, true) -- sleep xx seconds
 os.execute('sleep 30') -- Set lamp back to regular color and brightness
 brightness = 20 -- (0-100%)
 url = " --data-urlencode 'color={\"m\":2,\"t\":0,\"r\":0,\"g\":0,\"b\":0,\"cw\":0,\"ww\":255}' "
 url = url .. ' ' .. domoticzurl .. '/json.htm?type=command¶m=setcolbrightnessvalue&idx=' .. idx ..
 '&brightness=' .. brightness .. ''
 perform_webquery(url, true) -- Switch Off the light when required
 url = ' ' .. domoticzurl .. '/json.htm?type=command¶m=switchlight&idx=' .. idx .. '&switchcmd=Off'
 print(url) perform_webquery(url, true)
end
end

```

- force Notification for testing your changes

1. Open your "garbagecalendarconfig.lua" and set these debugging variables to true:
  - mydebug = true -- (true/false)
  - testnotification = true -- (true/false)
 This will force the first found event notification each run.

## We're done

You should be ready to go now. Ensure you reset the mydebug to:

```
"mydebug = false -- (true/false)"
```

```
testnotification = false -- (true/false)
```

to avoid a lot of messages in the Domoticz log.

check the [testing](#) page in case you need to test/debug your installation.

## Testing / Debugging

In case you have issues with your setup you can do the following steps described below.

When you still can't figure out the issue you have, you can also zip up the below defined **log & data** files and your **garbagecalendarconfig.lua** file and Email them to my GitHub Email address so I can check what is happening. :)

### 1. Check first your logfiles.

- Check these files exist in the directory your defined in garbagecalendarconfig.lua in variable **datafilepath** whether they exist and/or contain errors:
  - **\*\*garbagecalendar\_modulename.data\*\*** => the file containing the result of the webupdate process by the module.
  - **\*\*garbagecalendar\_modulename\_runlog\*\*** => log of the last time the script was ran
  - **\*\*garbagecalendar\_modulename\_run\_update.log\*\*** => log of the last time the script ran the update process in foreground
  - **\*\*garbagecalendar\_modulename\_run\_webupdate\_background.log\*\*** => log of the background webupdate process.
  - **\*\*garbagecalendar\_modulename\_run\_webupdate.log\*\*** => log of foreground process that shelled the background webupdate process.
- In case any of these is missing:
  - Check the **domoticz log** for any errors.
  - Check your directory name and the security/access for this directory, in case the files do not exist.
  - Set any/all of these debugging options to true to test the specified part of the logic:

```
-- Switch on mydebug in case of issues and initially and check the domoticz log for any issues or missing
-- set to true or false
mydebug = false -- run the script as it normally does when any of the scheduled times is the current
 time
testdataload = false -- run the web update module in the foreground with each run for debugging.
testdataloadbatch = false -- run the web update module in the background with each run for debugging.
 (testdataload takes presedence)
testnotification = false -- trigger a test notification each run for the first record for testing the
 notification system
```

- Also check **\*\*garbage.modulename.data\*\*** for valid table information looking like this format:

```
return {
 -- Table: {1}
 { {2}, {3}, },
 -- Table: {2}
 { ["garbagetype"]="pmd", ["wdesc"]="Plastic, Metalen en Drankkartons", ["garbagedate"]="2020-02-11", },
 -- Table: {3}
 { ["garbagetype"]="gft", ["wdesc"]="Groente, Fruit en Tuinafval", ["garbagedate"]="2020-02-13", },
}
```

## Raspberry PI with Debian Buster+

The default for tls is changed to TLSv1.2 which could give some issue on some sites not supporting this version yet. In case there is an issue retrieving the data for your GarbageCalendar you could try updating file **/etc/ssl/openssl.cnf** to:

```
[system_default_sect]
#MinProtocol = TLSv1.2
MinProtocol = TLSv1.0
```

- .. and reboot the system to activate this setting.
- **When the datafile is available and looking good, then check the LOG files in detail**

**\*\*garbagecalendar\_modulename\_run.log\*\***, **\*\*garbagecalendar\_modulename\_run\_\_update.log\*\*** and **Domoticz log** for any messages in case the text devices isn't updated. Normally it should say something like this in case the new text is the same as the current text (Domoticz text device will still be updated so you can see that the script has ran)

```
: ==> found schedule:Plastic, Metalen en Drankkartons: Din 11 feb ; Groente, Fruit en Tuinafval: Don 13 feb ;
Plastic, Metalen en Drankkartons: Din 18 feb ;
: No updated text for TxtDevice.
```

... and this when there is a update to be made for the text device:

```
: ==> found schedule:Plastic, Metalen en Drankkartons: Din 11 feb ; Groente, Fruit en Tuinafval: Don 13 feb ;
Plastic, Metalen en Drankkartons: Din 18 feb ;
: Update device from:
Groente, Fruit en Tuinafval: Don 6 feb
Plastic, Metalen en Drankkartons: Din 11 feb
Groente, Fruit en Tuinafval: Don 13 feb
replace with:
Plastic, Metalen en Drankkartons: Din 11 feb
Groente, Fruit en Tuinafval: Don 13 feb
Plastic, Metalen en Drankkartons: Din 18 feb
```

When the Domoticz device name is invalid you see something like this:

```
: Error: Couldn't get the current data of Domoticz text device: Container
```

## Notification issues

1. Ensure a record for each garbagetype is added to the garbagetype\_cfg table. The log will contain a Warning section when you are missing any record in this table. (see Setup page section 4.iv)
2. Ensure you defined an appropriate notification service configured and tested in Domoticz, as defined in Setup paragraph 5.
3. Switch testnotification on to force the testing of the notifications for the first type each minute:

```
testnotification = false -- (true/false) -- this will trigger a test notification for the first record for
testing the notification system
```

additional checking steps will be added as people start using it and issues are encountered. :-)