

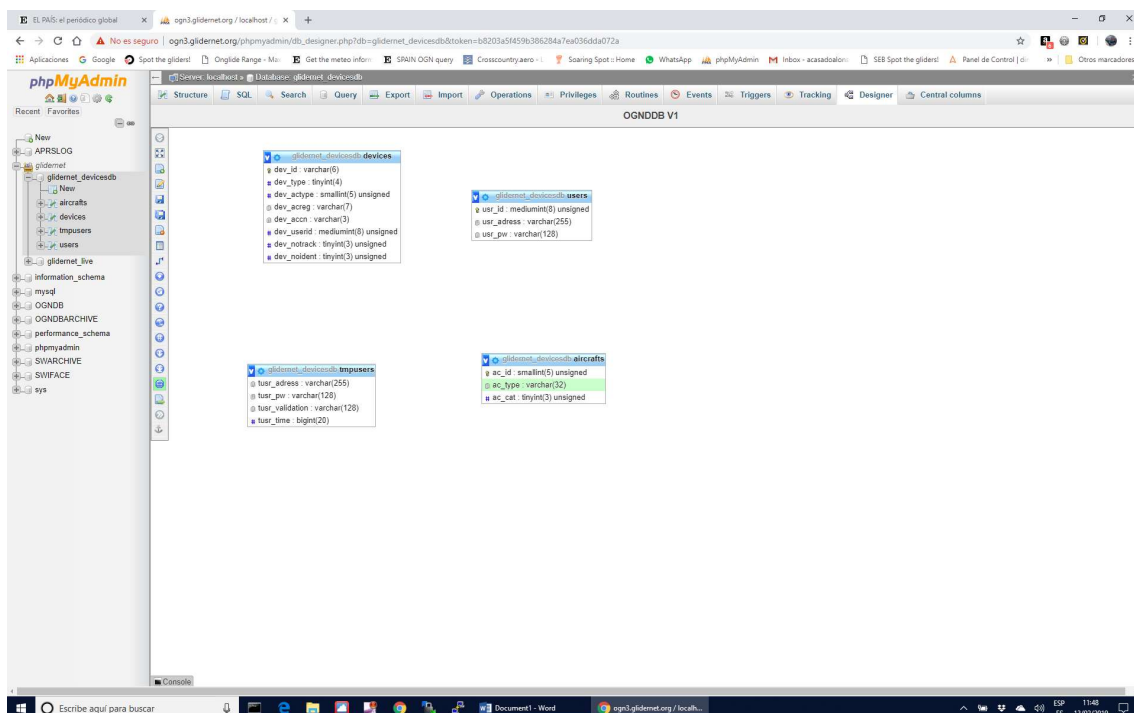
OGN Device Data Base (DDB) V2 proposal

Background:

The current OGN DDB contains the registration data of all the devices that can be handled by the OGN.

As version 1 the DB it contains 4 tables:

- *Devices*: that contains the registration information
- *Aircrafts*: is a table with all the possible aircraft types
- *Users*: with information about the registered users
- *TmpUsers*: with information about users that still did not confirm the email.



Rational for the change:

When the OGN DDB was designed, it was with the intention of register mostly Flarms and OGN trackers, under the assumption that only one device was in one plane. It was an alternate of FlarmNET.

Nowadays, we have a plethora of devices that pilots carry on their aircraft or paraglider, for example: Flarms, OGN trackers, Spider, Spot, InReach, Captur, Naviter Oudie, Flymaster, FANET devices, tablets with XCsoar, smart phones with apps like Naviter Navigator or XC Guide, etc., ...

The OGN is just not longer only for gliders and glider pilots, it is been used now more and more by paragliders, helicopters, tow planes, drones, etc.

Also there are many requests to be used by a plethora of virtual radars in conjunction with the popularity of drones.

And in many cases, the pilots can carry more than one device on board, but in that case, we do not want to show on the web apps as two or three aircrafts on the web map.

Also in the original we had a conflict with the FLARM devices registered with the ICAO ID or with the Internal ID, we had them as two different device types, when in reality is only one type: FLARM. So in the new design we had added a new field to the device table called dev_idtype where we identify the class of ID, either Internal or ICAO, in the future we can add more.

Proposal (V2.04):

What is proposed in this new version of the OGNDDDB, it is to decouple the information about the device information itself of the about the aircraft or flying object (glider, paraglider, helicopter, etc., ...).

So on the new design of the database, we will have 6 tables:

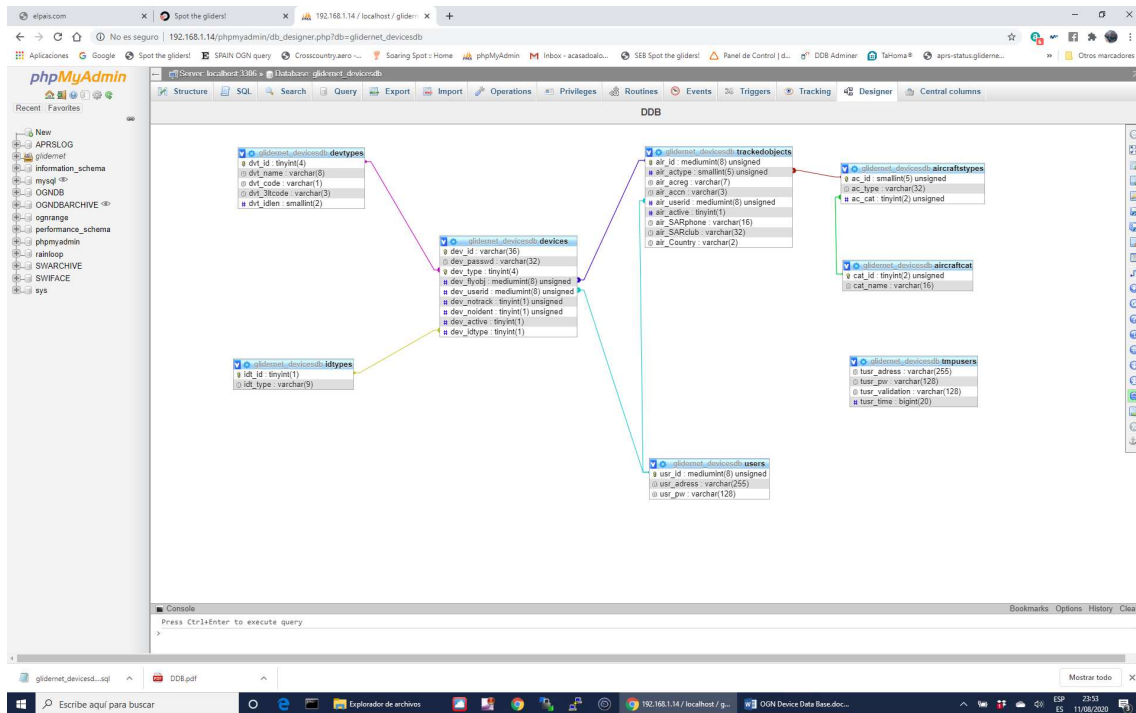
- DeviceTypes: a small table making the correspondence for the device type names: Flarms, OGNtrackers, SPOT, Naviter, InReach, Spider, Flymaster, etc. ..., and their assigned unique device type. It also contains information about how long is the

ID for this kind of device, the one char code and the 3 char code used on APRS.

- **Devices:** that contains the device registration information, but only info about the device, but with a link where this device is being carry on. We are adding the option of this device to be active or not. When the aircraft/tracked object is deleted, all the devices associated to that aircraft are deleted. We expanded the size of this field to be 36 chars, so we can accomplish other kind of device type like SPOT, Spider, ... also now each device type has its own address space.
- **IDTypes:** We used to identify on the devices what kind of ID uses, either internal or ICAO.
- **AircraftsTypes:** is a table with all the possible aircraft types. (same except of change the name to be: AircraftTypes).
- **AircraftCat:** a small table with all the aircraft type categories: Gliders, Planes, Ultralights, Drones, ...
- **Users:** with information about the registered users (same).
- **TmpUsers:** with information about users that still did not confirm the email (same). We need to have a script to delete the entries that are one month old.
- **TrackedObjects:** with information about the aircraft like registration, competition ID, aircraft type, etc. ... We are adding information that can be helpful in case of SAR like: phone, club and country(ISO). This information is optional. We are adding the option of this TrackedObject to be active or not.

See below the new design and the SQL data.

OGN DEVICE DATABASE



Filters

Containing the word

Table	Action	Rows	Type	Collation	Size	Overhead
aircraftcat	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	32 K	8
aircrafttypes	Browse Structure Search Insert Empty Drop	546	InnoDB	utf8_unicode_ci	39 K	8
devices	Browse Structure Search Insert Empty Drop	18,813	InnoDB	utf8_unicode_ci	2.2 MiB	-
devtypes	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8_general_ci	16 K	8
kitypes	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16 K	8
impusers	Browse Structure Search Insert Empty Drop	47	MyISAM	utf8_unicode_ci	6.2 K	8
trackedobjects	Browse Structure Search Insert Empty Drop	18,813	InnoDB	utf8_unicode_ci	2.2 MiB	-
users	Browse Structure Search Insert Empty Drop	5,711	InnoDB	utf8_unicode_ci	1.5 MiB	8
8 tables	Sum				47,952	InnoDB utf8_general_ci 6.1 MiB 8

Print Data dictionary

Create table

Name: Number of columns:

Go

Migration and API compatible:

Once that the new software has been tested, we can define a cut day and migrate with a specific utility the current data to the new format.

We have a small utility written in Python that does the migration of the structure and data from the old design DB to the new design DB. As we migrate the data we change the Flarms ICAO devices to be Flarms with IDtype ICAO. Also we scan for the last fix on the OGN systems and if we discover that the device has been used as ICAO we change the IDtype to be ICAO. We analyze as well the FANET and NAVITER devices and set the proper devices types.

In terms of the current API, we will maintain the compatibility, however we will extend it in order to gather the new data, perhaps restricting the new data to the JSON format for easier handling.

We have written a utility in order to analyze the consistency of the database.

API-Endpoints

/API/V1

We are entertaining thoughts of having a new REST API with the proper authorization schema, in order to allow to authorized developers to add dynamically devices and aircrafts/tracked objects from their own programs, but for now we had implemented a simple schema as follow:

URL request:

http://DDB.glidernet.org/API/V1/?action=add&login=john@acme.org&password=psw123456&device_id=757899&device_type=O®istration=EC-ACA&cn=AC&acftype=Discus

Reply on JSON format if successful:

```
{ 'NumberObjects': '18556', 'NumberDevices': '18545', 'ValidUser' :
'john@acme.org', 'Action': 'add', 'DeviceId' : '757899', 'DeviceType' :
'3', 'DeviceIdType' : '1', 'Registration' : 'EC-ACA', 'cn' :
'AC', 'AircraftType' : '105', 'FlyobjMsg' : ' flyobj_inserted ', 'FlyobjID' :
'22394', 'DeviceID' : ' 757899', 'DeviceMsg': 'device_inserted' }
```

```
{
  "NumberObjects": "18556",
  "NumberDevices": "18545",
  "ValidUser": "john@acme.org",
  "Action": "add",
  "DeviceId": "757899",
  "DeviceType": "3",
  "DeviceIdType": "1",
  "Registration": "EC-ACA",
  "cn": "AC",
  "AircraftType": "105",
  "FlyobjMsg": " flyobj_inserted ",
  "FlyobjID": "22394",
  "DeviceID": " 757899",
  "DeviceMsg": "device_inserted"
}
```

The possible actions are:

- **Object:** To add or update a tracked object. For add set the FlyObj=0
- **Delobject:** To delete a tracked object.
- **Device:** To add or update a tracked device. The FlyObj should belong to the same user.
- **Deldevice:** To delete a tracked device.
- **Add:** To add and tracked object and once created add a device and link it with that tracked object.

URL parameters

parameter	values	default	effect
action	Add,Object,Device,Delobejct,Deldevice	n/a	The action to execute
flyobj	decimal	0	ID of the tracked object

parameter	values	default	effect
device_id	Hex value	n/a	Select the device ID
device_type	F O N ...	n/a	select the device type ID
acftype	id	n/a	select aircraft type ID
registration	char	n/a	select registration
cn	char	n/a	select cn (comp ID)

API-Endpoints

/download

```
#DEVICE_TYPE,DEVICE_ID,AIRCRAFT_MODEL,REGISTRATION,CN,TRACKED,IDENTIFIED,IDTYPE,DEVACTIV
E,ACFTACTIVE
'F','000000','HPH 304CZ-17','OK-7777','KN','Y','Y','Internal','Y','Y'
'F','000002','LS-6 18','OY-XRG','G2','Y','Y','Internal','Y','Y'
'F','00000D','Ka-8','D-1749','W5','Y','Y','ICAO','Y','Y'
'F','000010','Unknown','D-EEAC','AC','Y','Y','Internal','Y','Y'
'F','000011','Skylane Airlony','D-MTEW','EW','Y','Y','Internal','Y','Y'
'F','000013','PA-28','D-EZIP','IP','Y','Y','Internal','Y','Y'
'O','000015','Paraglider','36445','','Y','Y','Internal','Y','Y'
'F','0000FD','Taurus','F-JRDN','DN','Y','Y','Internal','Y','Y'
```

URL parameters

parameter	values	default	effect
t	0/1	0	show field aircraft type if set to 1
j	0/1/2	0	forces JSON output when set to 1
device_id	csv	n/a	select a comma separated list of device ID's
from_id	id	n/a	select list of device ID's starting from this provided I
till_id	id	n/a	select list of device ID's until this provided ID

parameter	values	default	effect
registration	Csv	n/a	select a comma separated list of registrations
cn	Csv	n/a	select a comma separated list of call signs (comp ID

/download/?j=1

This returns all devices of the **DDB in JSON**. The output validates against the [ogn-ddb-schema-1.0.0](#). With **J=2** also returns as well the Device Types table, aircraft table and aircraft category table.

/download/download-fln.php

This returns the device database in a flarmnet-compatible format.

GitHub repository

The GitHub repo is at: <https://github.com/glidernet/ogn-ddb>

WEB presentation.

The screenshot displays the OGN Devices DataBase web application. The header shows the site name and a count of 19401 registered tracked devices. The sidebar on the left contains several management buttons: 'ADD T DEVICE AND T OBJECT', 'ADD TRACKED DEVICE', 'ADD TRACKED OBJECT', 'CLAIM OWNERSHIP', 'CHANGE PASSWORD', and 'DISCONNECT'. The main table lists tracked devices with columns for Device type, Device ID, ID type, Link to Regist., Link to CN, Tracking, Ident., and Active. The table shows several rows of data, including devices with IDs like 09cncBIVVFEcsbxqddrCfRbMqMPLau, 232418, 29C622, 2ADD43, 359C45, 3D417C, 5AC4E8, 5AC4EC, 5C0545, 60E6A0, 647654, 74D948, 88E1FB, and 88F308. The bottom part of the screenshot shows a second table with 18537 registered tracked objects, listing columns for Tracked Object, Aircraft type, Link to Regist., Link to CN, Active, Telephone, Club name, and Country. This table shows rows for objects like 1668, 8592, 8594, 8597, 8609, 8611, 8612, and 8614.

Device type	Device ID	ID type	Link to Regist.	Link to CN	Tracking	Ident.	Active
SPOT	09cncBIVVFEcsbxqddrCfRbMqMPLau	INTERNAL	N77UK	UK	✓	✓	✓
FLARM	232418	ICAD	N77UK	UK	✓	✓	✓
OGNT	29C622	INTERNAL	EC-AC1	AC1	✓	✓	✓
OGNT	2ADD43	INTERNAL	EC-AC2	AC2	✓	✓	✓
OGNT	359C45	INTERNAL	EC-AC3	AC3	✓	✓	✓
OGNT	3D417C	INTERNAL	EC-ACL	ACL	✓	✓	✓
OGNT	5AC4E8	INTERNAL	EC-ACP	ACP	✓	✓	✓
OGNT	5AC4EC	INTERNAL	EC-ACQ	ACQ	✓	✓	✓
OGNT	5C0545	INTERNAL	EC-ACO	ACO	✓	✓	✓
OGNT	60E6A0	INTERNAL	EC-ACN	ACN	✓	✓	✓
OGNT	647654	INTERNAL	EC-UFO	FBS	✓	✓	✓
OGNT	74D948	INTERNAL	EC-ACC	ACC	✓	✓	✓
OGNT	88E1FB	INTERNAL	EC-ACJ	ACJ	✓	✓	✓
OGNT	88F308	INTERNAL	EC-ACJ	ACJ	✓	✓	✓

Tracked Object	Aircraft type	Link to Regist.	Link to CN	Active	Telephone	Club name	Country
1668	ASW-20	LV-DRG	RJ	✓		Albatros	ARG
8592	SZD-48-1 Janitar Std 2	LV-EOM	PV	✓			
8594	ASW-20	LV-EPT	IR	✓			
8597	ASW-20	LV-EDK	CQ	✓			
8609	ASW-20	LV-ENV	CR	✓			
8611	SZD-41 Janitar Std	LV-DPG	PR	✓			
8612	ASW-20	LV-EQB	WR	✓			
8614	ASW-25	LV-EOF	ET	✓			

On the current presentation, we need to delete the aircraft data from the device information page and move it to a new push down page (**ADD TRACKED OBJECT**) similar to the current **ADD TRACKED DEVICE**

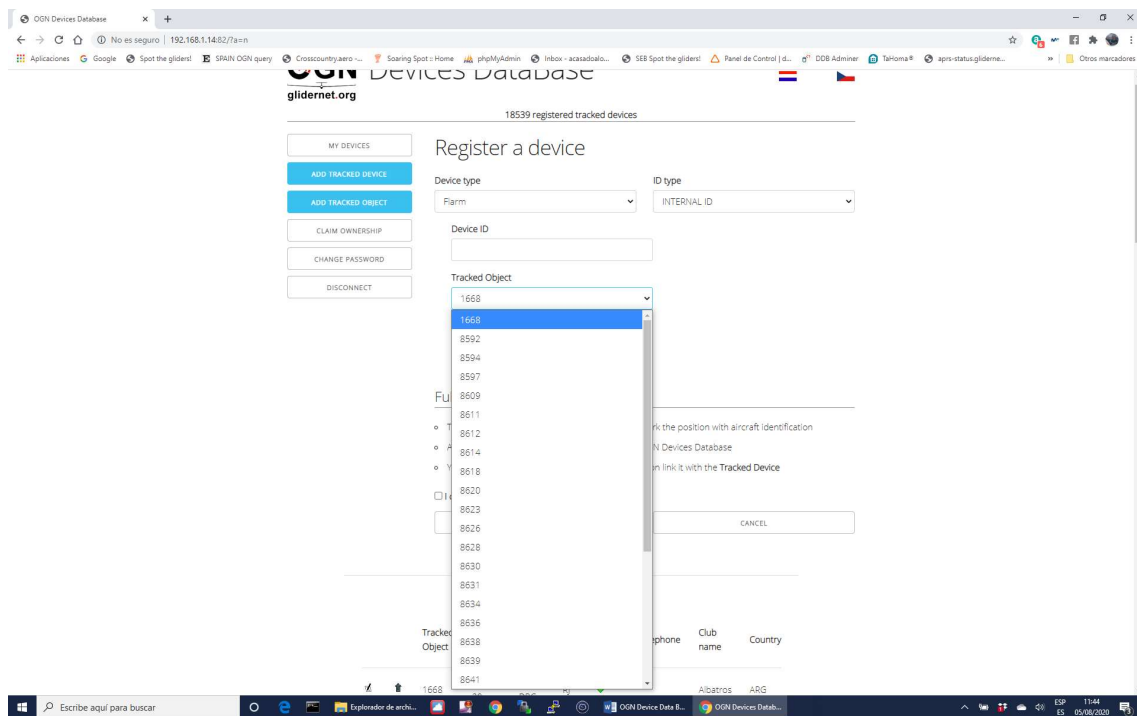
However for compatibility with the old versions and for simple users that just need to add a single device for a single aircraft we kept the option of add a device and object at the same time.

The screenshot shows the OGN Devices DataBase web application. The header includes the logo and the text "19401 registered tracked devices". The main content area is titled "Register a device and a tracked object". It contains a form with the following fields and options:

- Device type:** A dropdown menu with "Flarm" selected.
- ID type:** A dropdown menu with "INTERNAL ID" selected.
- Device ID:** A text input field.
- Aircraft type:** A dropdown menu with "A380" selected.
- Registration:** A text input field.
- Competition Number:** A text input field.
- Options:** Two checkboxes: "I don't want this device to be identified" and "I don't want this device to be tracked".
- Full participation:** A section with three bullet points:
 - Tracking applications that use the OGN DDB will mark the position with aircraft identification
 - Aircraft registration and CN are published in the OGN Devices Database
 - You need to create first a Tracked Object and later on link it with the Tracked Device
- Submit:** Two buttons: "SUBMIT" and "CANCEL".

At the bottom of the form, it says "19308 registered tracked objects". The browser's address bar shows "No es seguro | acasado.es/60082/ta=da". The taskbar at the bottom shows various application icons and the system clock indicating "11:03 26/10/2020".

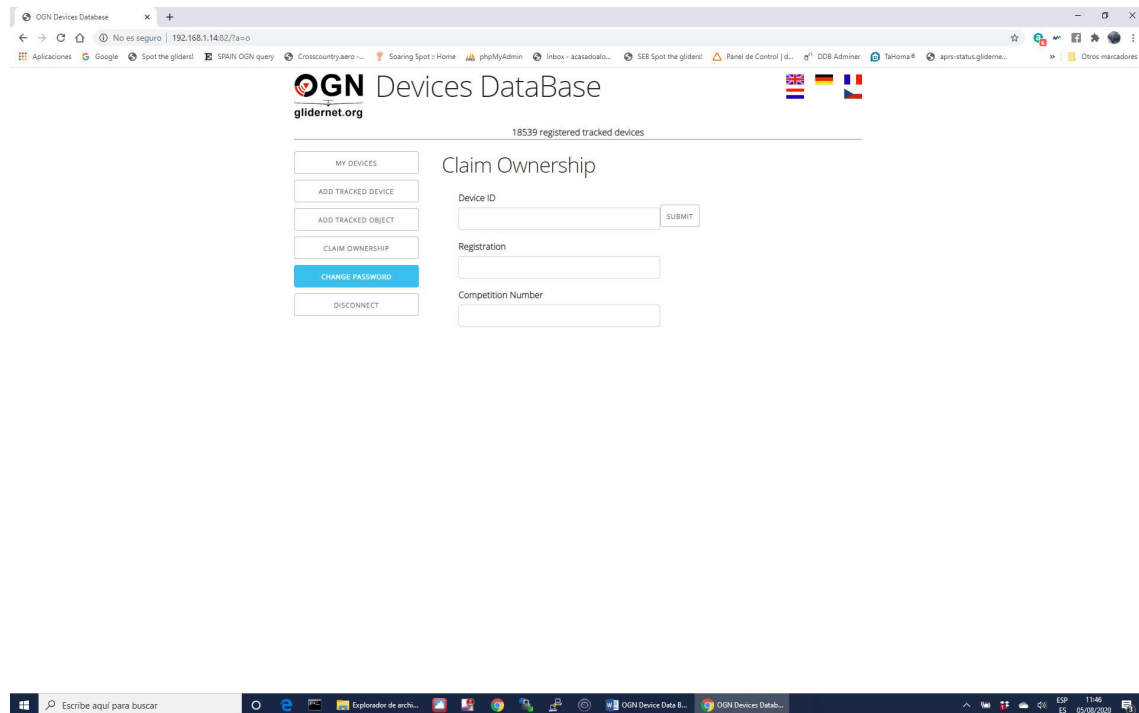
On the new screen we link the device with the tracked object in which list is showed on the low part of the screen.



OGNDDDB New feature:

We are adding a new feature to claim the ownership of a device, that is the situation where a person that registered a device, sell the device/aircraft and forgets to de register the device on the OGNDDDB, so the new feature CLAIM OWNERSHIP allows to send an email (hidden to the person claiming the ownership) to the person that registered that device asking to de register.

The person claiming the ownership should know: the device id, the registration and the competition ID



OGNDDDB Tables schema:



DDB nd



ogn-ddb-schema-2.0.json

OGNDDDB SQL



database schema v2.0