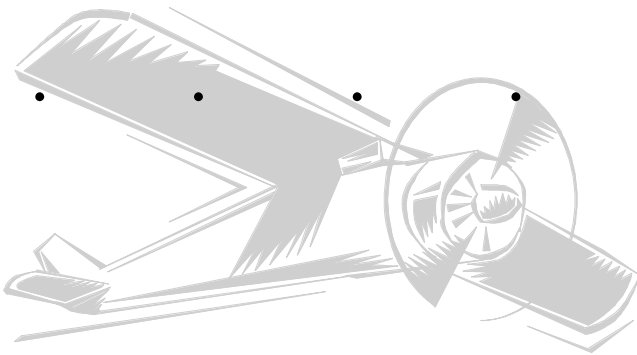


.....

CoPilot

# Waypoint Database



*Laurie J. Davis*

*ldavis@ieee.org*

*Version 4*



# CoPilot Waypoint Database

This document describes the format of the waypoint database used by the CoPilot flight planning program. It is intended for users that wish to create their own databases.

## Database Header

The waypoint database is a standard PDB file as described in the following document from Palm:

Palm® File Format Specification – document 3008-003  
<<http://www.palmos.com/dev/support/docs>>

The PDB header must be created with the following values:

```
name      = "CoPilot Waypoint"  
attributes = 8  
version   = 4  
type      = 'wayp' (32 bit integer) = 2002876784 decimal  
creator   = 'GXBU' (32 bit integer) = 1196966485 decimal
```

The PDB Record List contains records that point to the waypoint records. Each entry has a `uniqueId` that must be created.

If you set this `uniqueID` to zero, it will be set to a unique value when the database is loaded into the Palm device. This works fine, unless the user wants to replace your database with another version at a later time. When the user loads a newer version of the database, a new set of `uniqueIDs` will be assigned to the waypoint records. Any routes that have been set up will now refer to waypoints that no longer exist (the route database tracks waypoints via their `uniqueIDs`). The routes will essentially be erased. This method of assigning `uniqueIDs` is fine as long as you are willing to lose all of the route information for existing flights.

The other option is to assign the `uniqueIDs`. If you do this, you must ensure that the next time that you create a waypoint database you assign the same `uniqueIDs` to the waypoints.

If you wish the database to augment the existing database rather than replace it, you will need to create a database with a type of 'swpu' (1937207413) rather than 'wayp'. The name that you assign to the database can be anything except "CoPilot Waypoint". CoPilot will delete this database as soon as it has added the records to the currently installed database. If a record has a `uniqueID` that matches a record in the currently installed database, the existing record will be replaced. If a record has a `uniqueID` of zero or one that is not found in the currently installed database, the record will be added to the existing database.

The waypoint database file is normally a .pdb file. If you would like the file to be automatically copied directly to an external memory card, set the file extension to .way.

|           |        | File extension            |  |
|-----------|--------|---------------------------|--|
|           |        | .pdb                      | .way   |
| File Type | 'wayp' | Replace internal database | Create new external database or replace existing external database |
|           | 'swpu' | Add to internal database  | Not supported  |

## Waypoint Records

The structure of the individual records in the database is as follows:

```
struct WaypointStruct
{
    double    latitude;    // latitude in radians (north is positive)
    double    longitude;   // longitude in radians (west is positive)
    float     magVar       // magnetic variation in radians (west is positive)
    float     elevation;   // elevation in feet (even if other units are used for display)
    // the rest of the record is a set of zero terminated strings
    // the strings are packed together (i.e. do not pad the strings to the maximum length)
    // the strings MUST NOT be longer than the maximum size specified
    char      ident[];     // string for ident (max 10 characters PLUS the zero)
    char      name[];      // string for waypoint name (max 100 characters PLUS the zero)
    char      notes[];     // string for notes (max 1000 characters PLUS the zero)
};
```

The records in the database **MUST** be sorted alphabetically by ident. The sort order should ignore both case and accents (i.e. upper case, lower case, and accented characters are all considered the same). The ident must also be unique. If you have several waypoints with the same id, you will have to distinguish them in some way (e.g. id-1, id-2).

This format is valid for version 4.0 and later of CoPilot.

## AppInfo Block

You would expect that the creationDate field of the database header would contain the database creation date, but unfortunately this field gets reset to the date when the file was transferred to the Palm device. In order to overcome this, you can put the database creation date into the optional appInfo block. The information in this appInfo block is displayed via the “Waypoint Info ...” menu item. If an appInfo block is not provided, the creation date displayed is the date that the database was first loaded into the Palm device (creationDate from the database header).

The format of the appInfo block is as follows:

```
struct AppInfoStruct
{
    uint16    version;     // version of appInfo (set to zero)
    uint32    creationDate; // date to be displayed as the creation date
    char      dbInfo[];    // zero terminated string to be displayed on the form
};
```

The dbInfo string will be displayed on the “Waypoint Database” information form. You can use this string to provide any information that you wish.