

Records:

Always cool if you can make structs for our records as small as possible since you don't have a lot of memory. To make struct's more efficient, you can use < 8 bits per field Example:

```
Typedef struct
{
    unsigned field1: 4;
    unsigned flag: 1;
    unsigned stuff: 3;
} MyPackedType
MyPackedType rec;
Rec.field1 // deference like normal
```

Records in PalmOS

Palm maintains records in a DB in a sorted order.

Palm uses a call back to a sort function you provide to do sorting.

Example:

```
Int MyCompareFunc(MyRecType *rec1, MyRecType *rec2, Int16 other,
SortInfoPtr rec1sort, SortInfoPtr rec2Sort, memHandle AppInfo);
```

Intuitively, it should return a number < 0 if rec1 < rec2

a number > 0 if rec2 > rec1

if rec1==rec2 return 0

Beside rec1 and rec2, it is up to you to figure out how to use additional parameters passed to your function.

other –might be used to send what field to sort on

rec1sort/rec2sort - have two field UInt18 attribute (could be flags concerning record), UInt8 uniqueID[3] (uniqueID for rec)

To create a new record or find an existing record, need to find its sort position in DB

Example:

Database:

```
| rec1 | rec2 | ... | recn | delrec1 | delrec2 | ... | records marked for deletion
```

need to figure out where records should go

To find sort position

```
UInt16 DmFindSortPosition(DmOpenref dbP, SortRecordInfoPtr, void *newRec,
newRecordInfo, DmCompareF *compare, Int16 other)
```

(index of sort position) (record want to find position of) (can pass NULL if dont use)
(set to 0 if dont use) (cast your compare function to this type)

To create a record:

```
MyType rec;
// initialize it
```

```

....
index = DmFindSortPosition(gDB, &rec, NULL, (DmCompareF *)
    MyCompareFunc, NULL);
newRecordH=DmNewRecord(gDB, &index, sizeof(rec)); // memHandle
newRecordP = NewHandleLock(newRecordH); // memPtr
DmWrite( newRecordP, 0, &rec, sizeof(rec)); // offset from ptr value
MemHandleUnlock(newRecordH);
DmReleaseRecord(gDB, index, true); // set dirty bit

```

To read a record, can use DmFindSortPosition to find index of record you want (create a dummy rec with sort field filled in to do this search).

Then use:

```

    memHandle h =DmQueryRecord(gDB index); // then lock read data

```

To delete a record, use:

```

    DmDeleteRecod(gDB, index);

```

Modifying Records:

// first, set record

```

    MyType temp;
    recordH = DmGetRecord(gDB, index); // DmfindSortPosition
    record = MemHandleLock(recordH);
    temp = *record; //copy data
    temp.field = val; // modify value
    DmWrite(record, 0, &temp, sizeof(temp)); // wrote it back to database, but might now be
                                                out of sortorder

```

Use DmFindSortPosition to find new sort position

Use DmMoveRecord(gDB, index, newlocation);

```

    MemHandleUnlock(recordH);

```

```

    DmreleaseRecord(gDB, index, true);

```