



How to Use the Call History Database

Application Note AN-22

Revision 1.0

Document No. 80-0209-00

Table of Contents

1. Overview.....	1
2. How to Use	1
2.1. <i>The Call History Record</i>	<i>1</i>
2.2. <i>Accessing the Database.....</i>	<i>2</i>

1. Overview

The Phone application that comes with the VisorPhone™ Springboard™ Module or the Handspring Treo™ has a call history database built into it. The Phone application uses this database to store all incoming, outgoing, and missed calls. However, other applications can use the information stored in this database to track phone usage. For example, an application can keep track of the minutes used each day to call specific contacts. Or, it can track the number of incoming calls from other contacts. Or, it can keep track of the number of minutes left in the month. The actual usage of this information is left up to the imagination of the developer.

2. How to Use

The call history database can be used by any application. This database contains call records for each call that is placed or received by the Phone application. The actual call record is created when the connection is terminated. At that point, a call history record will be created that gives information about the call. This information includes: date and time of call, duration of call, origination or destination of call, and type of call.

2.1. The Call History Record

The record information is kept in a structure called CallHistoryRecordType.

```
typedef struct {
    UInt16      flags;
    DateType date;
    TimeType time;
    UInt32      duration;
    char*       name;
    char*       number;
} CallHistoryRecordType;
```

The flags field contains any specific flags that are specific to this call. At this time, there is only one flag defined.

```
#define callFlagRoaming    0x0001
```

If this bit is set, then this call was made while roaming.

The date field contains the date of the call, in `DateType` format.

The time field contains the time of the call, in `TimeType` format.

The duration field contains the duration of the phone call, in seconds.

The name field contains a pointer to the name of the person who was called, if it was found in the user's address book. If it was not found in the address book then it uses the Caller ID information and if that was not available it is NULL.

The number field contains a pointer to the phone number that was called. As the name field, it uses the address book entry, or the Caller ID information or it is NULL.

There are three types of calls that are kept in this database.

```
#define callCategoryIncoming    1
#define callCategoryMissed     2
#define callCategoryOutgoing    3
```

This information, however, is not kept in the actual call history record. It is stored in the record's attributes. This is because when these records are displayed, they can be filtered to display records of only a specific type of call.

2.2. Accessing the Database

To retrieve records from this database, you must create a function that grabs the data from the database, and fills in the record. The Call history Database type is "call" and the Creator ID is "HsPh". Here is an example routine on how to get and use the database.

```
Err CallDBGetRecord(DmOpenRef dbP, UInt16 index, CallHistoryRecordType* record,
    MemHandle* handleP)
{
    MemHandle handle;
    char* ptr;

    // Find the record in the database
    handle = DmQueryRecord (dbP, index);
    if (DmGetLastError())
    {
        *handleP = 0;
        return DmGetLastError();
    }
    // lock this record so we can access it
```

```
ptr = MemHandleLock (handle);

// Unpack the record
record ->flags = *(UInt16*)ptr;
ptr += sizeof(record->flags);
DateToInt(record->date) = *(UInt16*)ptr;
ptr += sizeof(record->date);
TimeToInt(record->time) = *(UInt16*)ptr;
ptr += sizeof(record->time);
record ->duration = *(UInt32*)ptr;
ptr += sizeof(record->duration);
record->name = ptr;
ptr += StrLen (ptr) + 1;
record->number = ptr;
*handleP = handle;

return 0;
}
```

Once you have this routine, getting call history data from the database is easy. Here is an example:

```
#define phoneCallDBType    'call'
#define hsFileCPhone      'HsPh'

DmOpenRef                  dbP;
MemHandle                  recordH;
CallHistoryRecordType     record;
UInt16                     recordNum;

// first, open the call history database (only need readOnly)
dbP = DmOpenDatabaseByTypeCreator(phoneCallDBType, hsFileCPhone,
                                  dmModeReadOnly);

// now, get the record from the call history database
CallDBGetRecord(dbP, recordNum, &record, &recordH);

// use the data in the call history record
.
// and we MUST unlock the handle when done
MemHandleUnlock(recordH);

// and finally, close the database
DmCloseDatabase(dbP);
```

Date	Revision #	Description of changes
9/7/2001	1.0	Initial release

Handspring™, Visor™, VisorPhone™, Springboard™, Treo™ and the Handspring and Springboard logos are trademarks or registered trademarks of Handspring, Inc. © 2001 Handspring, Inc.