



Palm OS[®] File Formats

Exploring Palm OS[®]

CONTRIBUTORS

Written by Denise Stone.

Engineering contributions by Ludovic Ferrandis, Keith Rollin.

Copyright © 1996–2004, PalmSource, Inc. and its affiliates. All rights reserved. This technical documentation contains confidential and proprietary information of PalmSource, Inc. (“PalmSource”), and is provided to the licensee (“you”) under the terms of a Nondisclosure Agreement, Product Development Kit license, Software Development Kit license or similar agreement between you and PalmSource. You must use commercially reasonable efforts to maintain the confidentiality of this technical documentation. You may print and copy this technical documentation solely for the permitted uses specified in your agreement with PalmSource. In addition, you may make up to two (2) copies of this technical documentation for archival and backup purposes. All copies of this technical documentation remain the property of PalmSource, and you agree to return or destroy them at PalmSource’s written request. Except for the foregoing or as authorized in your agreement with PalmSource, you may not copy or distribute any part of this technical documentation in any form or by any means without express written consent from PalmSource, Inc., and you may not modify this technical documentation or make any derivative work of it (such as a translation, localization, transformation or adaptation) without express written consent from PalmSource.

PalmSource, Inc. reserves the right to revise this technical documentation from time to time, and is not obligated to notify you of any revisions.

THIS TECHNICAL DOCUMENTATION IS PROVIDED ON AN “AS IS” BASIS. NEITHER PALMSOURCE NOR ITS SUPPLIERS MAKES, AND EACH OF THEM EXPRESSLY EXCLUDES AND DISCLAIMS TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, ANY REPRESENTATIONS OR WARRANTIES REGARDING THIS TECHNICAL DOCUMENTATION, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY WARRANTIES IMPLIED BY ANY COURSE OF DEALING OR COURSE OF PERFORMANCE AND ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, ACCURACY, AND SATISFACTORY QUALITY. PALMSOURCE AND ITS SUPPLIERS MAKE NO REPRESENTATIONS OR WARRANTIES THAT THIS TECHNICAL DOCUMENTATION IS FREE OF ERRORS OR IS SUITABLE FOR YOUR USE. TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, PALMSOURCE, INC. ALSO EXCLUDES FOR ITSELF AND ITS SUPPLIERS ANY LIABILITY, WHETHER BASED IN CONTRACT OR TORT (INCLUDING NEGLIGENCE), FOR DIRECT, INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, EXEMPLARY OR PUNITIVE DAMAGES OF ANY KIND ARISING OUT OF OR IN ANY WAY RELATED TO THIS TECHNICAL DOCUMENTATION, INCLUDING WITHOUT LIMITATION DAMAGES FOR LOST REVENUE OR PROFITS, LOST BUSINESS, LOST GOODWILL, LOST INFORMATION OR DATA, BUSINESS INTERRUPTION, SERVICES STOPPAGE, IMPAIRMENT OF OTHER GOODS, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR OTHER FINANCIAL LOSS, EVEN IF PALMSOURCE, INC. OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR IF SUCH DAMAGES COULD HAVE BEEN REASONABLY FORESEEN.

PalmSource, Palm OS, Palm Powered, HotSync, and certain other trademarks and logos are trademarks or registered trademarks of PalmSource, Inc. or its affiliates in the United States, France, Germany, Japan, the United Kingdom, and other countries. These marks may not be used in connection with any product or service that does not belong to PalmSource, Inc. (except as expressly permitted by a license with PalmSource, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits PalmSource, Inc., its licensor, its subsidiaries, or affiliates. All other product and brand names may be trademarks or registered trademarks of their respective owners.

IF THIS TECHNICAL DOCUMENTATION IS PROVIDED ON A COMPACT DISC, THE SOFTWARE AND OTHER DOCUMENTATION ON THE COMPACT DISC ARE SUBJECT TO THE LICENSE AGREEMENTS ACCOMPANYING THE SOFTWARE AND OTHER DOCUMENTATION.

Exploring Palm OS: Palm OS File Formats
Document Number 3120-002
November 9, 2004
For the latest version of this document, visit
<http://www.palmos.com/dev/support/docs/>.

PalmSource, Inc.
1240 Crossman Avenue
Sunnyvale, CA 94089
USA
www.palmsource.com

Table of Contents

About This Document	v
The <i>Exploring Palm OS</i> Seriesv
Additional Resources	vi
 1 Introduction to File Formats	 1
About the File Format Types2
File Formats Versus Memory Formats2
Palm Database (PDB) Files.2
Palm Resource (PRC) Files.3
Data Structures3
About Records and Resources3
About Database Formats.3
The Palm Database Header5
Palm Database Header Structure7
The Record List.9
Palm Database Record List Structure	10
About Multiple Record or Resource Lists in a Database	11
The Application and Sort Information Blocks	12
About Third Party Tools	12
 2 PDB and PRC Database Formats	 13
Overview of PDB and PRC Databases	14
Record and Resource Entries	15
PDB Record Entries.	15
PRC Resource Entry Fields	16
The Application Information Block	16
Finding the Length of the Application Information Block . . .	17
Standard Category Data in an Application Information Block	17
The Sort Information Block	19
Finding the Length of the Sort Information Block	19
PDB and PRC Raw Data	19
Reading and Writing PDB and PRC Data	20

3 Address Book Schemas	21
Home Information	22
Business Information	23
Other Information	24
Address Book Information	26
Locale Information	26
Application Information Block	27
4 Date Book Schemas	29
5 Memo Pad Schemas	31
6 To Do List Schemas	33
Index	35

About This Document

This book documents the two file formats that are commonly used in the Palm OS® platform. It also documents the schema formats for the PalmSource PIM applications: Address Book, Date Book, Memo Pad, and To Do List.

The *Exploring Palm OS* Series

This book is a part of the *Exploring Palm OS* series. Together, the books in this series document and explain how to use the APIs exposed to third-party developers by the fully ARM-native versions of Palm OS, beginning with Palm OS Cobalt. Each of the books in the *Exploring Palm OS* series explains one aspect of the Palm operating system, and contains both conceptual and reference documentation for the pertinent technology.

IMPORTANT: The *Exploring Palm OS* series is intended for developers creating native applications for Palm OS Cobalt. If you are interested in developing applications that work through PACE and that also run on earlier Palm OS releases, read the latest versions of the *Palm OS Programmer's API Reference* and *Palm OS Programmer's Companion* instead.

As of this writing, the complete *Exploring Palm OS* series consists of the following titles:

- *Exploring Palm OS: Programming Basics*
- *Exploring Palm OS: Memory, Databases, and Files*
- *Exploring Palm OS: User Interface*
- *Exploring Palm OS: User Interface Guidelines* (coming soon)
- *Exploring Palm OS: System Management*
- *Exploring Palm OS: Text and Localization*
- *Exploring Palm OS: Input Services*
- *Exploring Palm OS: High-Level Communications*

About This Document

Additional Resources

- *Exploring Palm OS: Low-Level Communications*
- *Exploring Palm OS: Telephony and SMS*
- *Exploring Palm OS: Multimedia*
- *Exploring Palm OS: Security and Cryptography*
- *Exploring Palm OS: Creating a FEP (coming soon)*
- *Exploring Palm OS: Porting Applications to Palm OS Cobalt*
- *Exploring Palm OS: Palm OS File Formats*

Additional Resources

- Documentation
PalmSource publishes its latest versions of this and other documents for Palm OS developers at
<http://www.palmos.com/dev/support/docs/>
- Training
PalmSource and its partners host training classes for Palm OS developers. For topics and schedules, check
<http://www.palmos.com/dev/training>
- Knowledge Base
The Knowledge Base is a fast, web-based database of technical information. Search for frequently asked questions (FAQs), sample code, white papers, and development documentation at
<http://www.palmos.com/dev/support/kb/>

Introduction to File Formats

There are two types of file formats that are commonly used in the Palm OS® platform:

- Palm™ database (PDB)
- Palm resource (PRC)

Files with a .pdb extension are record databases. Files with a .prc extension are resource databases. Please note, however, that the filename and extension on the desktop do not determine the name or type of database created on the handheld. The database header information inside the file determines a database name and type.

NOTE: Resource databases contain resources, not records; however, in some places the documentation and structure types use the term record generically to refer to the individual data entities stored inside of databases, including resource databases.

This chapter provides an overview of the file format types, including the database header that is used for each format.

[Chapter 2, “PDB and PRC Database Formats,”](#) on page 13 takes an in-depth look at the PDB and PRC file formats, which are almost identical.

About the File Format Types

This section provides an introduction to the two file format types that are described in this book. Each file format type is stored as a database.

In general, a database contains header information and a sequential list of records or resources. In addition, each database can contain one or two pieces of free-form data whose format is defined by the application that created it. The records within a database are similarly structured with record header information and record data.

File Formats Versus Memory Formats

This book describes the format of Palm databases that are stored in files on desktop computers. When one of these databases is loaded into a Palm Powered™ handheld, the database is stored in memory in a format that is similar to, but different than the format described in this book. The in-memory format of Palm databases is subject to change and is not documented by PalmSource, Inc.

Databases are typically imported into handheld devices when a user performs a HotSync® operation that installs an application. When a database is imported into a Palm Powered handheld, the Palm OS converts the database into standard Memory Manager objects. The Memory Manager tracks the size of each record or resource, and thus adds memory overhead; this means that the size of a database on the device is larger than its size on the desktop computer.

NOTE: The databases stored in ROM on Palm Powered handhelds are stored in a memory format, not in the file formats described in this book.

Palm Database (PDB) Files

A PDB is a record database generally used to store data for an application.

Palm Resource (PRC) Files

A Palm resource file contains a different type of data (resources instead of records), but has an almost identical structure to a PDB file. Palm OS applications are resource databases. A Palm OS application contains code resources as well as user interface resource elements.

Data Structures

The objects in Palm Database files can be represented by C structures, which are described in the chapters that follow.

About Records and Resources

Records and resources are both blocks of memory that contain any data you want. The exact definition of a record or resource is up to the application. From a low-level perspective, the difference between records and resources is the size and contents of the header for each object.

Records and resources are used for different purposes:

- Records are used to store application data such as memos or address book entries.
- Resources are used to store the code and user interface objects for an application.

You can treat records and resources as ordered or unordered databases. You can use a callback function to sort record databases; however, you cannot sort resources on a device. You can compare two records to determine the order in which they belong; however, an index does not exist.

About Database Formats

Each database is stored in a file on a desktop computer in sequential format, as shown in [Figure 1.1](#) on page 4. The format of each database file is logically structured as shown in [Figure 1.2](#) on page 5.

Introduction to File Formats

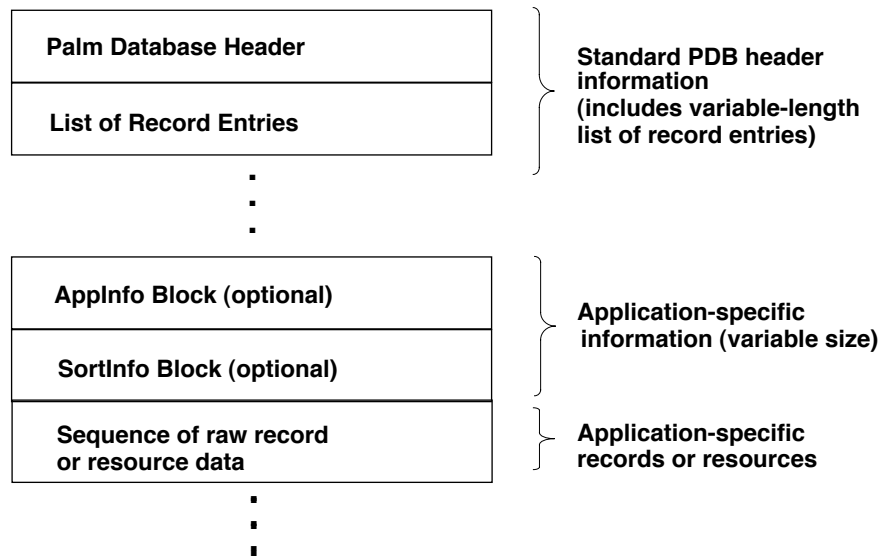
About Database Formats

Each database contains the following component parts:

- a database header that describes the database, references the `appInfo` and `sortInfo` blocks, and contains the record list, which references each record in the database
- an optional application information (`appInfo`) block in which you can store information specific to your application
- an optional sorting information (`sortInfo`) block in which you can store unique ID cross-reference tables or other meta information
- raw record or resource data

NOTE: All structure elements in all headers are byte-packed in network (big-endian) order.

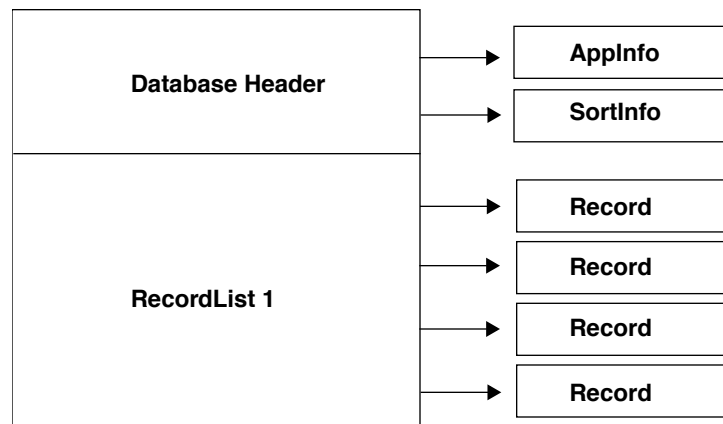
Figure 1.1 Database Storage Format



[Figure 1.2](#) on page 5 shows the logical representation of a record database file, with the header referencing the application information and sort information blocks, and with each record list referencing the raw data for the records stored in the database. The logical representation of a resource database file is the same, except that the record lists that refer to raw record data are replaced by resource lists that refer to raw resource data. The logical

representation of a web clipping application database is also very similar.

Figure 1.2 Logical Database Format for a Record Database



The Palm Database Header

The Palm database header is a standard `DatabaseHdrType` structure that is used to represent the header in PDB and PRC database files. The format of the header is shown in [Figure 1.3](#) on page 6. The byte values shown are offsets, in hexadecimal, from the beginning of the database (and of the header).

Introduction to File Formats

The Palm Database Header

Figure 1.3 Palm database header

byte	0	name	
		⋮	
	20	attributes	version
	24	creationDate	
	28	modificationDate	
	2C	lastBackupDate	
	30	modificationNumber	
	34	applInfoID	
	38	sortInfoID	
	3C	type	
	40	creator	
	44	uniqueIDSeed	
	48	recordList	
	4C	-----	

Note that the structure shown in [Figure 1.3](#) is how the header of a Palm Database is represented in a file on a desktop computer.

Palm Database Header Structure

The following structure represents a database file header:

Prototype

```
typedef struct {
    UInt8 name[dmDBNameLength];
    UInt16 attributes;
    UInt16 version;
    UInt32 creationDate;
    UInt32 modificationDate;
    UInt32 lastBackupDate;
    UInt32 modificationNumber;
    LocalID appInfoID;
    LocalID sortInfoID;
    UInt32 type;
    UInt32 creator;
    UInt32 uniqueIDSeed;
    RecordListType recordList;
} DatabaseHdrType;
```

Fields

name
A 32-byte long, null-terminated string containing the name of the database on the Palm Powered handheld. The name is restricted to 31 bytes in length, plus the terminator byte.

This name is also used to create the file name of the PDB when it is backed up during the HotSync process.

attributes
The attribute flags for the database.

version
The application-specific version of the database layout.

creationDate
The creation date of the database, specified as the number of seconds since 12:00 A.M. on January 1, 1970. (The defined base date is January 1, 1970, but for legacy reasons, it might be either January 1, 1900 or January 1, 1904 if the file was created on non-Windows platforms.)

modificationDate
The date of the most recent modification of the database, specified as the number of seconds since 12:00 A.M. on January 1, 1970. (The defined base date is January 1, 1970, but for legacy reasons, it might be either January 1, 1900 or

Introduction to File Formats

The Palm Database Header

January 1, 1904 if the file was created on non-Windows platforms.)

`lastBackupDate`

The date of the most recent backup of the database, specified as the number of seconds since 12:00 A.M. on January 1, 1970. (The defined base date is January 1, 1970, but for legacy reasons, it might be either January 1, 1900 or January 1, 1904 if the file was created on non-Windows platforms.)

`modificationNumber`

The modification number of the database.

`appInfoID`

The local offset from the beginning of the database header data to the start of the optional, application-specific `appInfo` block.

This value is set to NULL for databases that do not include an `appInfo` block.

`sortInfoID`

The local offset from the beginning of the PDB header data to the start of the optional, application-specific `sortInfo` block.

This value is set to NULL for databases that do not include a `sortInfo` block

`type`

The database type identifier.

For PDB databases, the value of this field depends on the creator application.

For PRC databases, this field usually has the value 'appl'.

`creator`

The database creator identifier.

`uniqueIDSeed`

Used internally by the Palm OS to generate unique identifiers for records on the Palm device when the database is loaded into the device.

For PRC databases, this value is normally not used and is set to 0.

`recordList`

A list of the records or resources in the database, as described in the next section.

IMPORTANT: There is always a gap between the final record list in the header and the first block of data in the database, where the first block might be one of the following: the `applInfo` block, the `sortInfo` block, raw record or resource data, or the end of the file. The gap is traditionally two bytes long; however, if you write code to parse a database, your code should be able to handle any size gap, from zero bytes long and up.

The Record List

The Palm database header ends with a record list. The record list has its own header, followed by 0 or more record entries. Each record entry describes a single record in the file.

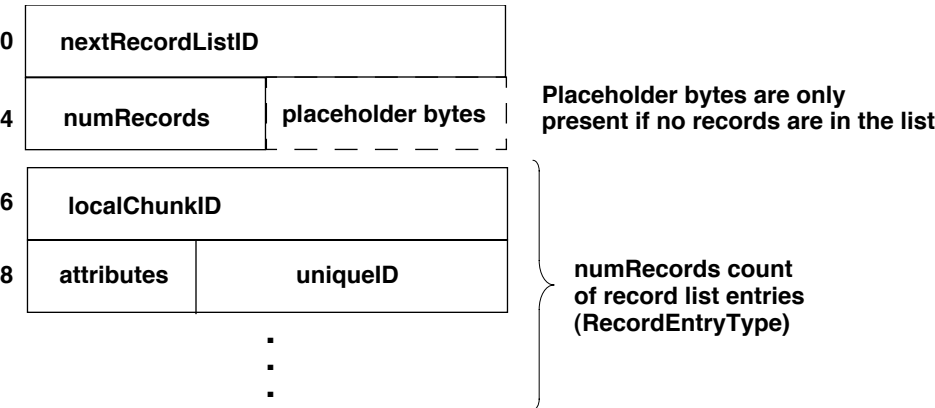
The record list has a variable length. When the database is loaded into a Palm Powered handheld, the Palm OS attempts to grow the list. If it cannot grow the list, the OS creates another record list and links it to the previous one by filling in the `nextRecordListID` field with the location of the new list. This capability is rarely used, and its use is discouraged by Palm. For more information, see “[About Multiple Record or Resource Lists in a Database](#)” on page 11.”

Each record entry references the location of the raw data for the record or resource and contains the attribute and ID information for that record or resource.

The remainder of this chapter describes the record list structure. However, the format of the record entries is different for different Palm database types. The record entry format for PDB databases and the resource entry format for PRC databases are shown in [Chapter 2, “PDB and PRC Database Formats.”](#)

[Figure 1.4](#) shows the structure of a record list.

Figure 1.4 Palm Database record list



Palm Database Record List Structure

The following structure declaration represents a Palm Database record list:

Prototype

```
typedef struct {
    LocalID nextRecordListID;
    UInt16 numRecords;
    UInt16 firstEntry;
} RecordListType;
```

NOTE: The placeholder bytes shown in [Figure 1.4](#) appear at the end of the record list, if there is one. If there is no list, these bytes appear just after the list header; otherwise, they appear after the last entry in the list.

Fields

`nextRecordListID`

The local chunk ID of the next record list in this database. This is 0 if there is no next record list, which is almost always the case.

For more information, see “[About Multiple Record or Resource Lists in a Database](#).”

`numRecords`

The number of record entries in this list.

`firstEntry`

The start of an array of record entry structures, each of which represents a single record in the list.

About Multiple Record or Resource Lists in a Database

The structure of Palm databases allows for multiple record lists in a single database; the record lists are chained together by setting the `nextRecordListID` field of the first record list to the offset of the next list in the database.

In practice, this capability is very rarely used, and the `nextRecordListID` field in the database header is almost always set to 0, which indicates that there is only one record list in the database. Since a single record list can be used to describe the maximum number of records (64K) in a file, multiple record lists are never required.

PalmSource, Inc. recommends against building databases with chained headers, and that your parsing code reject databases that have a non-zero value in the `nextRecordListID` field, to avoid potentially truncating such a database if your code encounters one.

A database with chained record lists might be encountered under very specific circumstances:

- when a huge database (one containing more than approximately 6000 records that has caused the headers to fragment) is beamed to a desktop OBEX stack from a Palm handheld device running version 3.5 or earlier of the Palm OS
- when code on a Palm handheld device uses the [`ExgDBWrite\(\)`](#) function to produce a PRB or PRC file image from such a database

NOTE: Version 4.0 and later of the Palm OS never produces chained record lists.

The Application and Sort Information Blocks

The database header can reference two optional application-specific blocks of information:

- The sort information (`sortInfo`) block
- The application information (`appInfo`) block

The `sortInfo` block is under your control. The OS does not use `sortInfo`. You can use it to store meta information about the database.

You are free to include whatever data you want in the `appInfo` block. However, there are restrictions on how you use this block if one of the following applies:

- your application uses Palm OS category functionality, as described in [Chapter 2, “PDB and PRC Database Formats,”](#) on page 13.

About Third Party Tools

There are a number of third party tools available for creating Palm databases on desktop computers, and for converting images in various formats into Palm image format. Rather than include a partial list, PalmSource, Inc. encourages you to search on the Internet for these tools, and recommends the following search terms:

- convert pdb
- convert prc

PDB and PRC Database Formats

This chapter describes the format of Palm OS[®] record (PDB) and resource (PRC) databases. Palm[™] record databases contain records that are used with applications that run on Palm Powered[™] handhelds. Palm resource databases contain application resources, including the code and the user interface objects for the application.

These databases are stored in memory on handheld devices, and are stored in file form on desktop computers. This chapter describes the file format of these databases, which is slightly different than their in-memory format. The in-memory format is subject to change and is not documented by PalmSource, Inc.

This chapter contains the following sections:

- [“Overview of PDB and PRC Databases”](#) on page 14 provides an overview of the database representation and shows an image of that representation.
- [“Record and Resource Entries”](#) on page 15 describes the entries that provide information about each record or resource in a database.
- [“The Application Information Block”](#) on page 16 describes the application information block that can optionally be included in PDB and PRC databases.
- [“The Sort Information Block”](#) on page 19 describes the sorting information block that can optionally be included in PDB and PRC databases.
- [“PDB and PRC Raw Data”](#) on page 19 describes how the raw record data is stored in PDB and PRC databases.
- [“Reading and Writing PDB and PRC Data”](#) on page 20 describes the Palm OS functions that you can use to convert a chunk of data to a Palm database, or convert a Palm database to a chunk of data.

PDB and PRC Database Formats

Overview of PDB and PRC Databases

For an overview of Palm databases and file formats, including a detailed description of the database header format, see [Chapter 1, “Introduction to File Formats.”](#)

NOTE: This chapter describes the format of PDB and PRC databases that are stored in files on desktop computers. When one of these databases is loaded into a Palm Powered handheld, the database is stored in memory in a format that is similar to, but different than the format described here. The in-memory format of PDBs and PRCs is subject to change and is not documented by PalmSource, Inc.

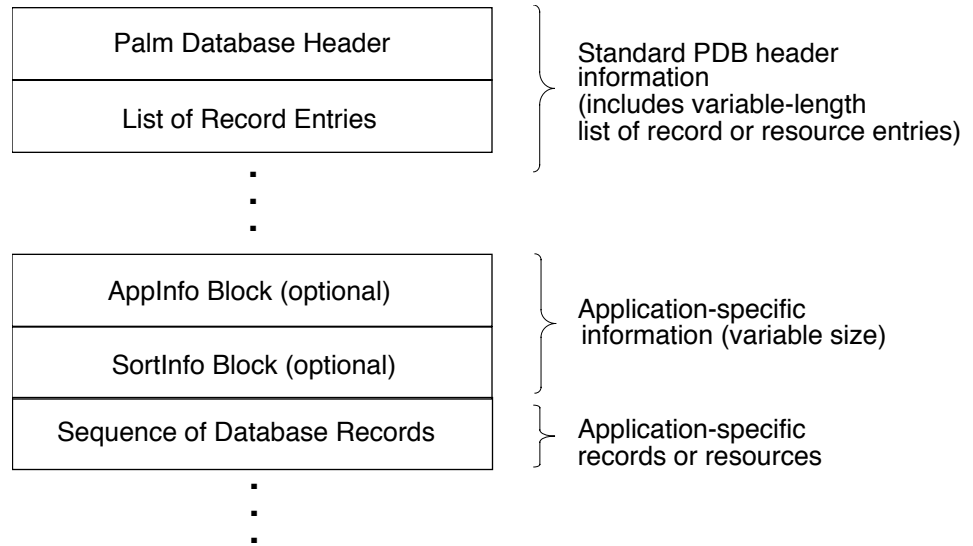
Overview of PDB and PRC Databases

Each PDB and PRC database contains the following components:

- A header containing fields that describe the database and refer to the information blocks and raw record data in the database. The Palm Database header is described in “[The Palm Database Header](#)” on page 5.
- A list of record entries, each of which describes a block of raw record or resource data.
- Two optional information blocks: the application information block and the sort information block.
- The raw record data, which is stored in linear format and referenced from the record list in the header.

[Figure 2.1](#) on page 15 shows the structure of a Palm database, as stored in a file on a desktop computer.

Figure 2.1 PDB and PRC database format



Record and Resource Entries

The record list in the Palm database header contains a list of entries that describe the raw data records or resources in the database. The record list is described in “[The Record List](#)” on page 9. The entries in PDB and PRC databases have different structures, and are described separately in this section.

PDB Record Entries

The following structure declaration represents a record entry in a PDB file:

```

Prototype  typedef struct {
                LocalID localChunkID;
                UInt8 attributes;
                UInt8 uniqueID[3];
            } RecordEntryType;

Fields     localChunkID
                The local offset from the top of the PDB to the start of the raw
                record data for this entry.

                Note that you can determine the size of each chunk of raw
                record data by subtracting the starting offset of the chunk
  
```

PDB and PRC Database Formats

The Application Information Block

from the starting offset of the following chunk. If the chunk is the last chunk, its end is determined by the end of the file.

attributes

Attributes of the record.

uniqueID

A three-byte long unique ID for the record.

PRC Resource Entry Fields

The following structure declaration represents a resource entry in a PRC file:

Prototype `typedef struct {
 UInt32 type;
 UInt16 id;
 LocalID localChunkID;
 } RsrcEntryType;`

Fields `type`

The resource type.

`id`

The ID of the resource.

`localChunkID`

The local offset from the top of the PRC to the start of the resource data for this entry.

Note that you can determine the size of each chunk of raw resource data by subtracting the starting offset of the chunk from the starting offset of the following chunk. If the chunk is the last chunk, its end is determined by the end of the file.

The Application Information Block

Each Palm Database can optionally include an application information (`appInfo`) block that contains arbitrary information.

The format of the `appInfo` block is determined by the creator of the database. However, PDBs that support the standard Palm OS category data, the `appInfo` block contains specific information, as described in “[Finding the Length of the Application Information Block](#),” below.

NOTE: PRCs can also contain application information blocks; however, this is rarely the case.

Finding the Length of the Application Information Block

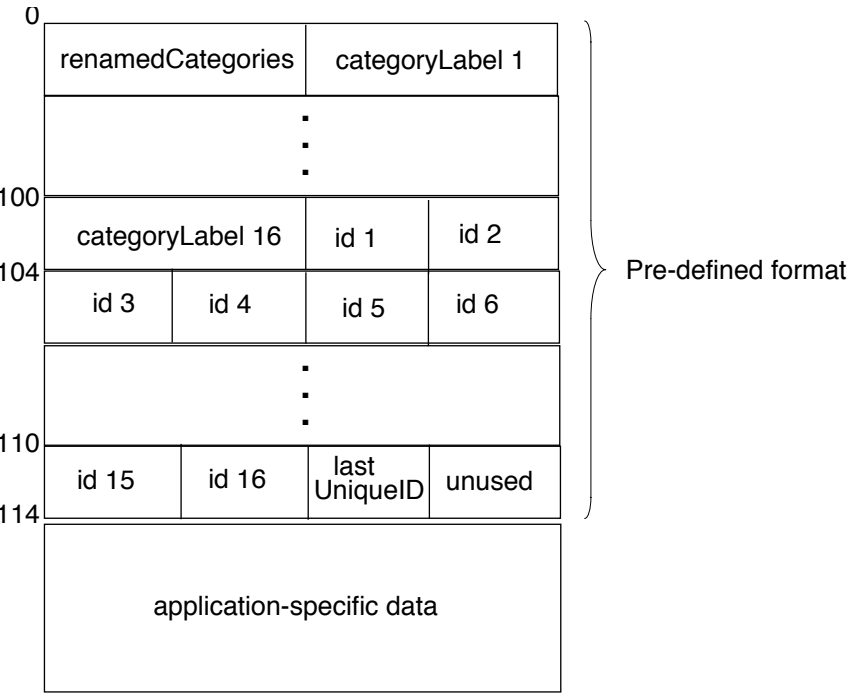
If the database includes an application information block, you can find its length by finding the block that follows it:

- If the database includes a sort information block, that block immediately follows the application information block.
- If the database does not include a sort information block, but does include one or more records, then the end of the application information block is just before the start of the first record block.
- If the database does not contain a sort information block and does not contain any records, then the end of the application information block is the end of the file.

Standard Category Data in an Application Information Block

A PDB for an application that supports standard Palm OS category data includes the category data in the standard format shown in [Figure 2.2](#) on page 18.

Figure 2.2 PDB applInfo bock for standard category data



The following structure declaration represents an application information block for an application that uses standard Palm OS category information:

Prototype

```
typedef struct {
    UInt16 renamedCategories;
    Char categoryLabels[16][16];
    UInt8 categoryUniqIDs[16];
    UInt8 lastUniqID;
    UInt8 padding;
} AppInfoType;
```

Fields

renamedCategories

Specifies which categories have been renamed.

categoryLabel

An array of 16 null-terminated category labels, each of which is 16 bytes long.

categoryUniqID

An array of 16 category ID values, each of which is one byte long.

`lastUniqID`

The last unique category ID assigned.

`padding`

Unused.

The Sort Information Block

The structure of the optional `sortInfo` block is completely up to the application; there is no standard format, nor is there a structure declared for this block. Most PDBs that contain a `sortInfo` block use it to store ordering information based on record IDs.

NOTE: PRCs can also contain sort information blocks; however, this is rarely the case.

Finding the Length of the Sort Information Block

If the database includes a sort information block, you can find its length by finding the block that follows it:

- If the database includes one or more records, then the end of the sort information block is just before the start of the first record block.
- If the database does not contain any records, then the end of the sort information block is the end of the file.

PDB and PRC Raw Data

Record data in a PDB is stored as a block of contiguous records. The local offset to the beginning of each record is stored in the record list(s) in the database header. The length and format of the record data is application-specific.

Similarly, resource data in a PRC database is stored as a block of contiguous resources. The local offset to the beginning of each resource is stored in the record list(s) in the database header. The length and format of the resource data is not documented in this book.

Reading and Writing PDB and PRC Data

The Palm OS provides functions that you can use to convert data into or out of Palm Database formats:

- If you have a chunk of data on the handheld device that is formatted as described in this chapter, you can use either the [`DmCreateDatabaseFromImage\(\)`](#) function or the [`ExgDBRead\(\)`](#) function to convert that data into a Palm Database.
- If you want to convert a Palm Database into a chunk of data on the handheld device, you can use the [`ExgDBWrite\(\)`](#) function.

Address Book Schemas

This chapter describes the format of the Address Book schemas. For more information on working with schema databases, see *Exploring Palm OS: Memory, Databases, and Files*.

The Address Book schema is split in 5 parts:

- Home information (Personal Group):
 - Reserved ID range: 100 - 2900
(see [Table 3.1](#) on page 22)
- Business information (Business Group):
 - Reserved ID range: 3000 - 5900
(see [Table 3.2](#) on page 23)
- Other information (Other Group):
 - Reserved ID range: 6000 - 8900
(see [Table 3.3](#) on page 25)
- Address Book information:
 - Reserved ID range: 9000 - 9900
(see [Table 3.4](#) on page 26)
- Locale information:
 - Reserved ID range: 10000 - 12000
(see [Table 3.5](#) on page 27)

Home Information

The Home information (Personal Group) of the AddressBookDBSchema database encompasses the reserved ID range of 100 - 2900.

The column ID ranges are defined in this way:

- Range 0 - 15999 reserved for PalmSource
- Range 16000 - 19999 reserved for developers
- Range 20000 - 32735 reserved for licensees

Subranges should be defined by licensees

Table 3.1 AddressBookDBSchema Home Information

Column Name	Col ID	Description	Type	Size
Title	100	Name Prefix	VarChar	256
Last Name	200	Last Name	VarChar	256
Middle Name	300	Middle Name	VarChar	256
First Name	400	First Name	VarChar	256
Suffix	500	Name Suffix	VarChar	256
Nickname	600	Nickname	VarChar	256
Profession	700	Profession of the person	VarChar	256
Phone	800	Home phone number	VarChar	256
Mobile	900	Personal mobile phone number	VarChar	256
Fax	1000	Personal fax number	VarChar	256
Email 1	1100	First personal email	VarChar	256
Email 2	1200	Second personal email	VarChar	256
Email 3	1300	Third personal email	VarChar	256
Street	1400	Personal address: Street Name	VarChar	256

Table 3.1 AddressBookDBSchema Home Information

Column Name	Col ID	Description	Type	Size
City	1500	Personal address: City Name	VarChar	256
State	1600	Personal address: State Name	VarChar	256
ZipCode	1700	Personal address: Zip code number	VarChar	256
Country	1800	Personal address: Country Name	VarChar	256
Anniversary	1900	Anniversary date	Date	-
Birthday	2000	Birthday of the person	Date	-

Business Information

The Business information (Business Group) of the AddressBookDBSchema database encompasses the reserved ID range of 3000 - 5900.

The column ID ranges are defined in this way:

- Range 0 - 15999 reserved for PalmSource
- Range 16000 - 19999 reserved for developers
- Range 20000 - 32735 reserved for licensees

Subranges should be defined by licensees

Table 3.2 AddressBookDBSchema Business Information

Column Name	Col ID	Description	Type	Size
Company	3000	Company Name	VarChar	256
Title	3100	Title/Role in the company	VarChar	256
Street	3200	Business address: Street Name	VarChar	256
City	3300	Business address: City Name	VarChar	256

Address Book Schemas

Other Information

Table 3.2 AddressBookDBSchema Business Information

Column Name	Col ID	Description	Type	Size
State	3400	Business address: State Name	VarChar	256
ZipCode	3500	Business address: Zip code number	VarChar	256
Country	3600	Business address: Country Name	VarChar	256
Phone	3700	Business phone number	VarChar	256
Mobile	3800	Business mobile phone number	VarChar	256
Fax	3900	Business fax number	VarChar	256
Email	4000	Business email address	VarChar	256
Assistant Name	4100	Assistant Name	VarChar	256
Assistant Phone	4200	Assistant phone number	VarChar	256

Other Information

The Other information (Other Group) of the AddressBookDBSchema database encompasses the reserved ID range of 6000 - 8900.

The column ID ranges are defined in this way:

- Range 0 - 15999 reserved for PalmSource
- Range 16000 - 19999 reserved for developers
- Range 20000 - 32735 reserved for licensees

Subranges should be defined by licensees

Table 3.3 AddressBookDBSchema Other Information

Column Name	Col ID	Description	Type	Size
Street	6000	Other address: Street Name	VarChar	256
City	6100	Other address: City Name	VarChar	256
State	6200	Other address: State Name	VarChar	256
ZipCode	6300	Other address: Zip code number	VarChar	256
Country	6400	Other address: Country Name	VarChar	256
Phone	6500	Other phone number	VarChar	256
Mobile	6600	Other mobile phone number	VarChar	256
Fax	6700	Other fax number	VarChar	256
Pager	6800	Pager number	VarChar	256
Custom 1	6900	Custom field 1	VarChar	256
Custom 2	7000	Custom field 2	VarChar	256
Custom 3	7100	Custom field 3	VarChar	256
Custom 4	7200	Custom field 4	VarChar	256
ICQ	7500	ICQ identifier	VarChar	256
AIM	7600	AIM identifier	VarChar	256
Yahoo	7700	Yahoo identifier	VarChar	256
MSN	7800	MSN identifier	VarChar	256
Jabber	7900	Jabber identifier	VarChar	256
URL	8500	URL	VarChar	256

Address Book Information

The Address Book information of the `AddressBookDBSchema` database encompasses the reserved ID range of 9000 - 9900.

The column ID ranges are defined in this way:

- Range 0 - 15999 reserved for PalmSource
- Range 16000 - 19999 reserved for developers
- Range 20000 - 32735 reserved for licensees

Subranges should be defined by licensees

Table 3.4 AddressBookDBSchema Address Book Information

Column Name	Col ID	Description	Type	Size
Note	9000	Note	VarChar	256
Display Phone	9100	Column ID of the default phone	UInt32	-
Last Viewed Tab ID	9200	Last visited tab	UInt32	-

Locale Information

The Locale information of the `AddressBookDBSchema` database encompasses the reserved ID range of 10000 - 12000.

The column ID ranges are defined in this way:

- Range 0 - 15999 reserved for PalmSource
- Range 16000 - 19999 reserved for developers
- Range 20000 - 32735 reserved for licensees

Subranges should be defined by licensees

Table 3.5 AddressBookDBSchema Locale Information

Column Name	Col ID	Description	Type	Size
Yomi Last Name	10000	Japanese Yomi Last Name	VarChar	256
Yomi First Name	10100	Japanese Yomi First Name	VarChar	256
Yomi Company	10200	Japanese Yomi Company Name	VarChar	256
English Name	11000	Chinese English Name	VarChar	256

Application Information Block

The AddressBookDBSchema database also has an appInfo block in which information specific to the Address Book is stored.

Table 3.6 AddressBookAppInfo

Column	Col ID	Description	Type	Size
Country	100	User choice of the device country	UInt16	-

Address Book Schemas

Application Information Block

Date Book Schemas

This chapter describes the format of the Date Book schema. For more information on working with schema databases, see *Exploring Palm OS: Memory, Databases, and Files*.

Table 4.1 DateBookDBSchema

Column Name	Col ID	Description	Type	Size
Repeating Event	10	True if this is a repeating event.	Boolean	-
Untimed Event	20	True if this is an untimed event. (The start time is then 12:00 am in device time zone.)	Boolean	-
Start date & time	30	START time & date in posix epoch time. (UTC seconds since 1/1/1970.)	Int32 *	-
End date & time	40	END time & date in posix epoch time. (UTC seconds since 1/1/1970.)	Int32 *	-
Timezone	50	Timezone name ("US/Central" for example)	VarChar	32
Time advance	100	Advance in next unit (if set to 0xFF => no advance set)	Int8	-
Time advance unit	110	Advance unit (AlarmUnitType)	UInt8	-
Repeat type	200	Among repeatNone, repeatDaily, repeatWeekly, repeatMonthlyByDay, repeatMonthlyByDate, repeatYearly	UInt8	-
Repeat end date	210	END repeat date in posix epoch time. (UTC seconds since 1/1/1970.)	Int32 *	-

Date Book Schemas

Table 4.1 DateBookDBSchema (continued)

Column Name	Col ID	Description	Type	Size
Repeat frequency	220	Frequency (for example, every 2 days if <code>repeatType</code> daily)	UInt8	-
Repeat On	230	<code>monthlyByDay</code> and <code>repeatWeekly</code> only (the days in week, or the day number in month)	UInt8	-
Repeat start of week	240	<code>repeatWeekly</code> only (Sunday / Monday)	UInt8	-
Exceptions dates	300	Exception dates in posix epoch time. (UTC seconds since 1/1/1970.) Only the day is extracted to get the exception day, the time in day is then discarded but is used for time zone computations.	UInt32 Vector	64Kb
Description string	400	Description	VarChar	256
Location string	410	Location	VarChar	32
Note string	420	Note	VarChar	64Kb

* Will be modified by the new [DateTimeType](#) column data type.

Memo Pad Schemas

This chapter describes the format of the Memo Pad schema. For more information on working with schema databases, see *Exploring Palm OS: Memory, Databases, and Files*.

Table 5.1 MemoDBSchema

Column Name	Col ID	Description	Type	Size
Memo	100	The memo text	VarChar	64Kb

Memo Pad Schemas

To Do List Schemas

This chapter describes the format of the To Do List schema. For more information on working with schema databases, see *Exploring Palm OS: Memory, Databases, and Files*.

Table 6.1 ToDoDBSchema

Column Name	Col ID	Description	Type	Size
Due Date	100	Due Date	Date	-
Priority	200	Priority	Int8	-
Description	300	Description	VarChar	64Kb
Note string	400	Note	VarChar	64Kb

To Do List Schemas

Index

A

- Address Book schemas 21
- AddressBookAppInfo 27
- AddressBookDBSchema 22, 23, 25, 26, 27
- appInfo. *See* application information block
- application information block 12
 - PDB 16
 - PRC 16

B

- byte packing 4

D

- database format
 - PDB 14
 - PRC 14
- DatabaseHdrType 5
- databases
 - byte packing in 4
 - gap in header 9
 - in file 2
 - in memory 2
 - logical format of 5
 - storage format of 4
 - with multiple record lists 11
- Date Book schema 29
- DateBookDBSchema 29

G

- gap bytes 9

H

- header
 - PDB 5
- HotSync 2

M

- Memo Pad schema 31
- MemoDBSchema 31

P

- Palm database 1
 - header structure 7

- types 1

- Palm database header 5
 - record list in 9

- Palm databases
 - and third party tools 12

- Palm resource 1

- PDB 1, 2
 - application information block 16
 - database format 14
 - header 5
 - raw record data 19
 - sort information block 19

- PQA
 - header 5

- PRC 1, 3
 - application information block 16
 - database format 14
 - header 5
 - raw record data 19
 - sort information block 19

R

- raw record data
 - PDB 19
 - PRC 19
- record database 13
- record list 9
 - multiple instances in database 11
 - structure 10
- Records
 - and resources 3
- resource database 13
- resources
 - and records 3

S

- schemas
 - Address Book 21
 - Date Book 29
 - Memo Pad 31
 - To Do List 33
- sort information block 12
 - PDB 19
 - PRC 19
- sortInfo. *See* sort information block

T

third party tools 12
To Do List schema 33
ToDoDBSchema 33