IPD File for BlackBerry

By lff0305@gmail.com

V0.1

Please get the latest version **from http://code.google.com/p/bbipd/**

Chapter 1 General Structure

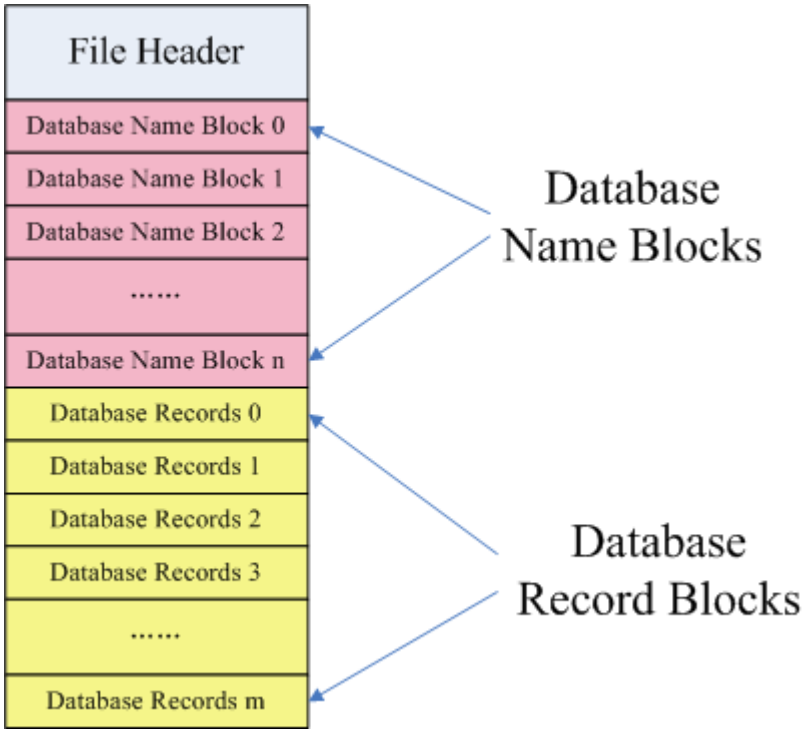
An IPD file can be considered as a database collection. When you backup data to an IPD file from RIM Blackberry Desktop Manager, data on the phone is saved into one or more databases. Each kind of data is saved to one database. For example, all SMS(Short Message Service) data is saved in Database named by SMS Messages.

The IPD file contains the following part:

The **file header** : The header of the IPD file. The signature, version data are in this part.

The **Database Name Blocks**: Several blocks containing the Database names.

The **Database Records**: Several records contain the real data.



Chapter 2 File Header

File Header is a small piece of data. It has the following data:

RIM Signature

LineBreak

Database Version

Numbers of Databases in current file

Database Separator

Name	Length	Offset	Description
RIM Signature	37 bytes	0x0	RIM signature:**The ASCII of Inter@ctive Pager Backup/Restore File**
LineBreak	1 byte	0x25	0x0A. Seems not used.
Database Version	1 byte	0x26	In recent versions of BlackBerry Desktop Manager this field is 02.
Numbers of Databases	2 byte	0x27~0x28	How many Databases are in this file
Database Separator	1 byte	0x29	0. Seems not used.

The following is an example from a real .IPD file.

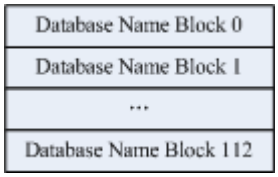
```
00000000h: 49 6E 74 65 72 40 63 74 69 76 65 20 50 61 67 65 ; Inter@ctive Page
00000010h: 72 20 42 61 63 6B 75 70 2F 52 65 73 74 6F 72 65 ; r Backup/Restore
00000020h: 20 46 69 6C 65 0A 02 00 71 00 15 00 56 69 73 75 ;  File...q...Visu
```

RIM signature | LineBreak | Version | Number of Databases | Database Seperator

Sample (1) File Header

Chapter 3 Database Name Blocks

Database Name blocks are after the Header part and they matched with DatabaseNumber value in Header part. For example, from the Sample(1), the number of Databases is 0x 00 71. So the numbers in current database will be 7 * 16 + 1 = 113:



Total 113 Database Name Blocks

In each block ,the following data is saved:

Name Length

Name (Including Terminating NULL)

Name	Length	Offset From start of Database Name Block	Description
Name Length	2 bytes	0x0	The length of the Database Name including NULL as **Little Endian**
Name	NA	0x2	The NAME and Terminating NULL

The following is an example from a real .IPD file.

```
00000000h: 49 6E 74 65 72 40 63 74 69 76 65 20 50 61 67 65 ; Inter@ctive Page
00000010h: 72 20 42 61 63 6B 75 70 2F 52 65 73 74 6F 72 65 ; r Backup/Restore
00000020h: 20 46 69 6C 65 0A 02 00 71 00 15 00 56 69 73 75 ;  File...q...Visu
00000030h: 61 6C 20 56 6F 69 63 65 20 4D 61 69 6C 20 3F 3F ; al Voice Mail ??
00000040h: 00 12 00 50 69 6E 79 69 6E 20 49 4D 20 6F 70 74 ; ...Pinyin IM opt
00000050h: 69 6F 6E 73 00 18 00 41 70 70 6C 69 63 61 74 69 ; ions...Applicati
00000060h: 6F 6E 20 50 65 72 6D 69 73 73 69 6F 6E 73 00 16 ; on Permissions..
                  15 00 : Name Length  56 69 : Name
```

Sample (2) Database Name Blocks

This example shows 3 Database Names:

(1) : Length = 0x 00 15 = 21 (In file it is 0x 15 00 and it is saved as Little Endian. So when converting to an int, it should be 0x 00 15)

Name = `Visual Voice Mail ??\0 (20 characters, 1 terminating NULL)`

(2) : Length = 0x 00 12 = 18

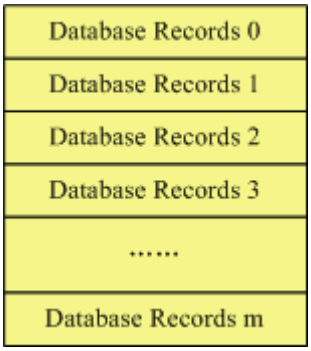
Name = `Pinyin IM options\0 (17 characters, 1 terminating NULL)`

(3) : Length = 0x 00 18 = 24

Name=`Application Permissions\0 (23 characters, 1 terminating NULL)`

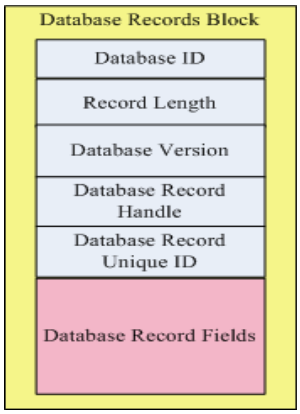
Chapter 4 Database Record Blocks

Database Record Blocks are after Database Name Blocks. Each ipd file contains one or more Database Record Blocks. It can be displayed as:



And each Database Record Block contains the following data:

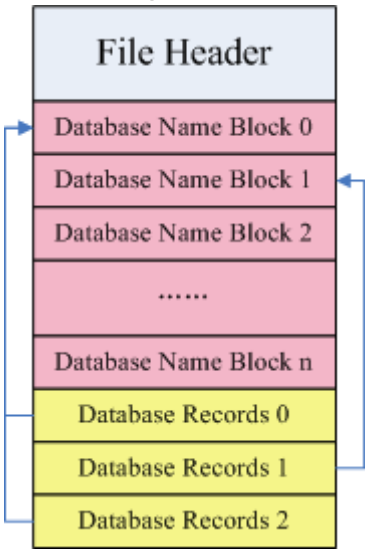
- Database ID
- Total Record Length
- Database Version
- Database Record Handle
- Database Record Unique ID
- Database Fields



A Database Record Block

Name	Length	Offset	Description
Database ID	2 bytes	0x00	An integer (Little Endian). The offset index of the Database Name Block to which this Database Record Block belongs. For example, if the Database ID is 0x 03 00 (3), then this record belongs to the Database which name is the 3rd in Database Name blocks.
Total Record Length	4 bytes	0x02	The total length(Little Endian) of this record. We will discuss about it later.
Database Version	1 byte	0x06	The version of the Database Record Format. Currently 0x05 is used.
Database Record Handle	2 bytes	0x07	An integer for the record handle.
Database Record Unique ID	4 bytes	0x08	An unique ID in the ipd file.
Database Fields	NA	0x0c	Fields in this record. Will discuss later.

Database ID: So as described from the above table, from, we can know to which database this record belongs:



This example shows:

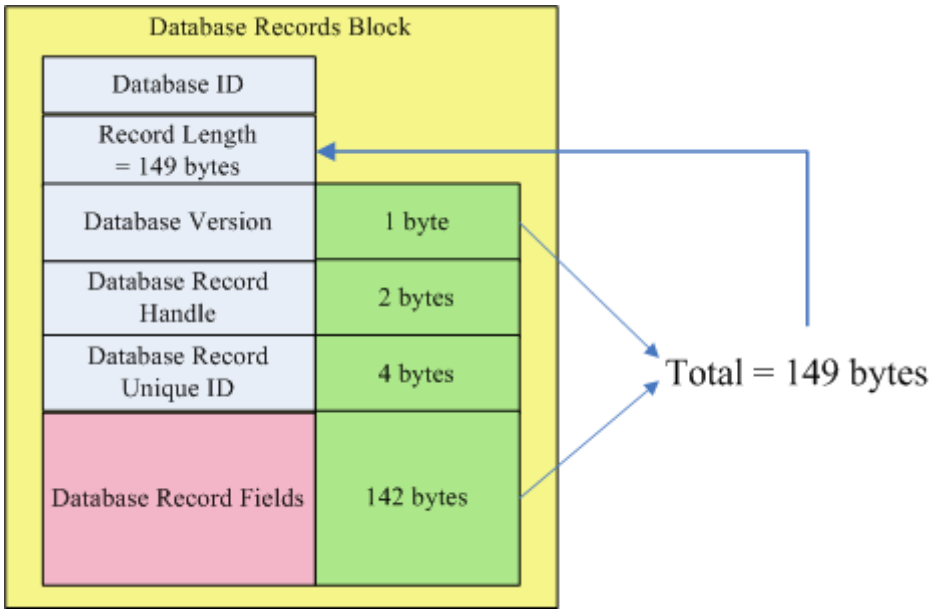
Database Record 0 ==> Database 0
Database Record 1 ==> Database 1
Database Record 2 ==> Database 0

Total Record Length: This field is a little complicated to understand. Let's get it from one example. Assume we have a database record with 7 fields (For detail data in field, we will discuss

later).

Field Number	Length
0	64
1	8
2	4
3	6
4	16
5	15
6	8
Total	121

In this example, the total length of the records is 121 bytes. And each record, has a field header of 3 bytes. So the total length is 121 + 3 * 7 = 142 bytes. What's more, the Database Version, Database Record Handle, Database Record Unique ID and Database Fields are all need to calculated. So the length is :142 + 4 + 2 + 1 = 149 bytes. And in ipd file, the length is saved as Little Endian as 4 bytes. So it is saved finally as 0x 95 00 00 00.



Record Length Calculation