

4JFile Component

Creating JFile™ Databases for PalmOS™ devices using 4th Dimension™

Developer Manual

Version 2.0

BaileyWare

670 S Clover Avenue
San Jose, CA 95128
(408) 398-1796

<http://www.BaileyWare.com>

Software License and Limited Warranty

PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THE SOFTWARE. BY USING THE SOFTWARE, YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE SOFTWARE LICENSE AND WARRANTY DISCLAIMER (collectively referred to herein as the "Agreement"). THIS AGREEMENT CONSTITUTES THE COMPLETE AGREEMENT BETWEEN YOU AND Gerald E. Bailey, known also as BaileyWare. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE AND PROMPTLY RETURN THE PACKAGE FOR A FULL REFUND OF THE PURCHASE PRICE, NOT INCLUDING SHIPPING OR HANDLING.

1. **Ownership of Software.** The enclosed manual and computer programs ("Software") were developed and are copyrighted by Gerald E. Bailey (BaileyWare) and are licensed, not sold, to you by BaileyWare for use under the following terms, and BaileyWare reserves any rights not expressly granted to you. You own the disk(s) on which any software is recorded, but BaileyWare retains ownership of all copies of the Software itself. Neither the manual nor the Software may be copied in whole or in part except as explicitly stated below.

2. **License.** BaileyWare, as Licensor, grants to you, the LICENSE, a non-exclusive, non-transferable right to use this Software subject to the terms of the license as described below:

i) You may make backup copies of the Software for your use provided they bear the BaileyWare copyright notice.

ii) You may use this Software in an unlimited number of custom or commercial databases or applications created by the original license. No additional product license or royalty is required.

3. **Restrictions.** You may not distribute copies of the Software to others (except as an integral part of a database or application within the terms of this License) or electronically transfer the Software from one computer to another over a network. You may not distribute copies of the Software as an integral part of a development shell.

4. **Termination.** This license is effective until terminated. This license will terminate immediately without notice from BaileyWare if you fail to comply with any of its provisions. Upon termination you must destroy the Software and all copies thereof, and you may terminate this license at any time by doing so.

5. **Update Policy.** BaileyWare may create, from time to time, updated versions of the Software. At its' option, BaileyWare will make such updates available to the Licensee.

6. **Warranty Disclaimer.** THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. BaileyWare DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIALS IN THE TERMS OF CORRECTIONS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. IF THE SOFTWARE OR WRITTEN MATERIALS ARE DEFECTIVE YOU, AND NOT BaileyWare OR ITS' DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES, ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY BaileyWare, IT'S DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY,

AND YOU MAY NOT RELY ON SUCH INFORMATION OR ADVICE. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

7. **Governing Law.** This agreement shall be governed by the laws of the State of California.

Copyright and Trademarks

All trade names referenced in this document are the trademark or registered trademark of their respective holder.

4JFile is copyright ©2006 Gerald E. Bailey (BaileyWare).

Palm, and PalmOS are registered trademarks of Palm, Inc.

4th DIMENSION, ACI, ACI US, 4D, 4D Server, 4D Client, and 4D Insider and 4D Compiler are trademarks of 4D SA and 4D, Inc.

Windows is a trademark of Microsoft Corporation.

Macintosh and Mac OS X are trademarks of Apple Computer, Inc.

Introduction:

BaileyWare is pleased to announce Version 2.0 of the 4JFile Component for 4th Dimension. 4JFile allows 4th Dimension developers to create, read and synchronize JFile type PalmOS databases directly from within 4th Dimension. This is accomplished using the following two commands:

- **Palm_Create_JFilePDB**
- **Palm_Read_JFilePDB**

JFile Version 5.x is a fast, efficient, and user friendly database program for the PalmOS Platform. For more information on JFile, please see the developers website: <http://www.land-j.com/>

JFile Key Features:

- Up to 50 Fields!
- Up to 32,000 Database Records!
- Up to 32,000 Database Records!
- NEW in 5.65: Support for the Treo navigation buttons, Tungsten T5 compatibility
- Added support for 320x480 mode on most OS 5 machines that support it (such as T3, UX50,etc)
- Allows for both portrait and landscape mode on units that support it

4JFile Key Features:

- Easy to use!
- Extremely Flexible!
- Only 1 command needed to create the JFile Palm database file!
- Only 1 command needed to read the JFile Palm database file!

Installing 4JFile Components

1. Start the 4D Insider Application
2. Open the database that you wish to install the component using 4D Insider.
3. Choose the Install/Update... command from the Components menu.
4. Select the component file and click the Open button.

4D Insider will automatically detect if the developer is attempting a new installation or an upgrade. If the component the developer is attempting to install is already installed, the version numbers will be compared and the developer will be asked whether to continue with the installation, or cancel the update.

Important!

Please note: The *Demonstration Version* of the 4JFile Component limits the creating or reading of JFile databases to only 5 records.

Upgrading User Notes:

This section is for users upgrading to Version 2.0 of the 4JFile Component from previous versions. Previous versions of the component contained different command names and syntax. Please modify your 4th Dimension code accordingly.

Command Changes:

Previous Version

4Palm_Create_JFilePDB

4Palm_Read_JFilePDB

Version 2.0

[Palm_Create_JFilePDB](#)

[Palm_Read_JFilePDB](#)

4JFile Command Syntax

LongInt:=**Palm_Create_JFilePDB**(fileName;->databaseOptionsArray;->arrayFieldNames;->arrayFieldTypes;
->arrayFieldAttributes;->arrayUniqueIDs;->dataPointerArrays)

Parameter	Type		Description
pathName	Text	→	The pathname of the PDB file to create
databaseOptionsArray	Pointer	→	Pointer to Longint array of Database Options
arrayFieldNames	Pointer	→	Pointer to Array of Field Labels
arrayFieldTypes	Pointer	→	Pointer to Array of Field Types
arrayFieldAttributes	Pointer	→	Pointer to Array of Field Attributes
arrayUniqueIDs	Pointer	←	Pointer to Array of Unique Record IDs
dataPointerArrays	Pointer	→	Pointer to Array of Pointers to Data Arrays
Result	LongInt	←	Error code returned (See section 4JFile Error Codes)

Description

The **Palm_Create_JFilePDB** is used to create JFile type PalmOS databases, also known as PDBs.

NOTE: All arrays are to be passed as pointers, otherwise an error will be generated.

pathName is the complete path to the file to be created and installed on the handheld device. This will most likely be something similar to the following:

Mac OS X:

Macintosh HD:Users:gbailey:Documents:Palm:Users:Gerald Bailey:Files to Install:myJFileDatabase.pdb

Windows XP:

C:Program Files:palmone:gbailey:Install:myJFileDatabase.pdb

Note: If the filename variable/parameter is set to "", the Save File dialog box will be displayed. It is not necessary to add the file extension ".PDB" to the filename parameter as the component will do this for you. Example: "MyDatabase" not "MyDatabase.PDB"

databaseOptionsArray is a 3 element LongInt array consisting of the following values:

Array{1} = Read Only or Read Write Database (0=Read Write, 1=Read Only)

Array{2} = Password Protected Database (0=Normal, 1=Password Required)

Array{3} = Locked Database Structure (0=Unlocked, 1=Locked)

arrayFieldNames is an alpha 20 array of field labels which will be displayed on the Palm. This needs to be the same size as the arrays contained within the dataPointerArrays array (see below). The array should be of type string with a length of 20.

arrayFieldTypes is a LongInt array of field types on the Palm. This needs to be the same size as the arrays contained within the dataPointerArrays array (see below). Acceptable field types are:

- 1 = String (also text)
- 2 = Boolean
- 4 = Date
- 32 = Time
- 8 = Long Integer
- 16 = Real
- 64 = Popup
- 70 = Calculated
- 73 = Date Popup - Creates Popup Date Picker
- 80 = Time Popup - Creates Popup Time Picker

arrayFieldAttributes is a LongInt array of field attributes. This needs to be the same size as the arrays contained within the dataPointerArrays array (see below). Acceptable field types are:

- 0 = Normal Field
- 1 = Read Only field
- 2 = Hidden Field

arrayUniqueIDs is a LongInt array of unique record ids on the Palm. This array will be sized and populated by the developer's successful execution of the [Palm_Create_JFilePDB](#) command. This information can be saved within the developers database in order to provide a method of finding records to synchronize.

dataPointerArrays is a pointer array, containing pointers to all the arrays (up to a maximum of 50) of data the developer wishes to send to the JFile database.

Example: Given the following arrays from the SELECTION TO ARRAY command:

```
SELECTION TO ARRAY([COMPANY]Company_Name;arr_CompanyName_a40;[COMPANY]Company_Phone;  
arr_CompanyPhone_a10;[more fields];more arrays...)
```

The **dataPointerArrays** array should be:

```
arr_DataArrays_ptr{1}:=>arr_CompanyName_a40  
arr_DataArrays_ptr{2}:=>arr_CompanyPhone_a10  
.  
.  
.  
arr_DataArrays_ptr{n}:=>nextArray...
```

LongInt:=**Palm_Read_JFilePDB**(\$FileName;->arrayInfo;->arrayData)

Parameter	Type		Description
pathName	Text	→	The pathname of the PDB file to read
arrayInfo	Pointer	←	Pointer to 2 Dimensional Array to hold column information
arrayData	Pointer	←	Pointer to 2 Dimensional Array to hold row (record) data
Result	LongInt	←	Error code returned (See section 4JFile Error Codes)

Description

The **Palm_Read_JFilePDB** is used to create JFile type PalmOS databases, also known as PDBs.

NOTE: All arrays are to be passed as pointers, otherwise an error will be generated.

pathName is the complete path to the file to be created and installed on the handheld device. This will most likely be something similar to the following:

Mac OS X:

Macintosh HD:Users:gbailey:Documents:Palm:Users:Gerald Bailey:Files to Install:myJFileDatabase.pdb

Windows XP:

C:Program Files:palmone:gbailey:Backup:myJFileDatabase.pdb

arrayInfo is a two dimensional array containing the column information such as field labels and field types.

Example:

```
arr_ColDef_t{1}{1}  First Field Label (ie: Company Name)
arr_ColDef_t{1}{2}  First Field Type (ie: "1" for String/Text)
arr_ColDef_t{2}{1}  Second Field Label (ie: Active Client)
arr_ColDef_t{2}{2}  Second Field Type (ie: "2" for Boolean)
arr_ColDef_t{3}{1}  Third Field Label (ie: Total Purchases)
arr_ColDef_t{3}{2}  Third Field Type (ie: "16" for Real)
.
.
.
arr_ColDef_t{n}{1}
arr_ColDef_t{n}{2}
```

Note: See the section labeled *JFile Data Types* for more information about valid field types.

arrayData is a two dimensional array containing the actual row data from the Palm device.

Example:

```
arr_RowDef_t{1}{1}  First Records' First Field of Data (ie: "Apple Computer, Inc.")
arr_RowDef_t{1}{2}  First Records' Second Field of Data (ie: "1 Infinite Loop")
arr_RowDef_t{2}{1}  Second Records' First Field of Data (ie: "4D, Inc.")
arr_RowDef_t{2}{2}  Second Records' Second Field of Data (ie: "3031 Tish Way")
.
.
.
arr_RowDef_t{n}{n}
arr_RowDef_t{n}{n}
```

Note: Element zero (ie: arr_RowDef_t{1}{0}) will contain each Palm records' unique ID.
The last element (ie: arr_RowDef_t{1}{n}) will contain each Palm records' status (ie: modified/deleted)

Examples:

Example use of the [4JFile_Create_JFilePDB](#) command:

```
C_STRING(31;$FileName)
```

```
$FileName:="" ` This will cause the Save File dialog to be displayed.
```

```
ARRAY LONGINT(databaseOptionsArray;3)
```

```
databaseOptionsArray{1}:=0 ` Read Write Database
```

```
databaseOptionsArray{2}:=0 ` Not Password Protected
```

```
databaseOptionsArray{3}:=1 ` Lock the structure to prevent the Palm user from modifying it. (Make syncing more reliable)
```

```
` Gather the data here.
```

```
ALL RECORDS([COMPANY])
```

```
SELECTION TO ARRAY([COMPANY]Company_Name;arr_CompanyName_a40;[COMPANY]Active_Status;
```

```
arr_CompanyStatus_b;[COMPANY]Gross_Sales;arr_Gross_Sales_r)
```

```
` Create and set up the field names to be displayed on the Palm device.
```

```
ARRAY STRING(20;arrayFieldNames;3) ` Assuming only 3 fields (arrays) will be used.
```

```
arrayFieldNames{1}:="Company"
```

```
arrayFieldNames{2}:="Active?"
```

```
arrayFieldNames{3}:="Gross Sales"
```

```
` Create and set up the field attributes here.
```

```
ARRAY LONGINT(arrayFieldAttributes;Size of array(arrayFieldNames))
```

```
arrayFieldAttributes{1}:="1" ` String/Text
```

```
arrayFieldAttributes{2}:="2" ` Boolean
```

```
arrayFieldAttributes{3}:="16" ` Real
```

```
` Create the array that will receive each Palm records' unique ID.
```

```
ARRAY LONGINT(arrayUniqueIDs;0)
```

```
` Create and set up the pointer array to be passed to the Component.
```

```
ARRAY POINTER(dataPointerArrays;Size of array(arrayFieldNames))
```

```
dataPointerArrays{1}:= ->arr_CompanyName_a40
```

```
dataPointerArrays{2}:= ->arr_CompanyStatus_b
```

```
dataPointerArrays{3}:= ->arr_Gross_Sales_r
```

```
` Create the JFile Palm Database here.
```

```
$Error_n:=Palm_Create_JFilePDB(FileName;->databaseOptionsArray;->arrayFieldNames;
```

```
->arrayFieldTypes;->arrayFieldAttributes;->arrayUniqueIDs;->dataPointerArrays)
```

Example use of the [4JFile_Read_JFilePDB](#) command:

```
C_STRING(31;$FileName)
```

```
$FileName:="" ` This will cause the Open File dialog to be displayed.
```

```
` Create and set up the two dimensional arrays to receive the JFile data.
```

```
ARRAY TEXT(arrayInfo;0;2)
```

```
ARRAY TEXT(arrayData;0;0)
```

```
` Read in the JFile Palm Database here.
```

```
$Error_n:=Palm\_Read\_JFilePDB($FileName;->arrayInfo;->arrayData)
```

Frequently Asked Questions: (And other useful tips)

Q) I have a client who has a 4D database that I developed for them. They would like to download data stored in their 4D database such as inventory, customer history, etc to his Palm device. Will 4JFile do this for me?

A) Yes, 4JFile is the perfect solution for your customer.

Q) Will I need to purchase a copy of the 4JFile component for every database I wish to deploy with it installed?

A) Yes, you must purchase a license for each database you deploy with the 4JFile Component installed.

Q) If I'm deploying on 4D Server, with both Macintosh and a Windows clients, must I purchase a version for each platform?

A) Yes, when you deploy a 4D Server solution with the 4JFile Component installed, you must purchase a license for each platform you are deploying on. In the case above, you are required to purchase a Macintosh and Windows license for each deployed database of this type.

Q) If I'm deploying an application merged with 4D Engine, do you offer volume licensing?

A) Yes. Contact BaileyWare for more information on our OEM Program.

Q) I understand that there are conduits which allow data to be synched with PalmOS devices. Is that how this works?

A) No, 4JFile is not a conduit and does not actually perform the syncing operations. You can however perform logical syncing of data yourself using the data that is provided by the component such as unique record numbers and record status.

Q) Does the PalmOS device have to be in the cradle when I use the commands included with the component?

A) Since the 4JFile component does not perform the actual syncing, the handheld device does not have to be in the cradle until you're ready to HotSync.

Q) If 4JFile isn't a conduit, how so I get the JFile database I created onto my Palm device?

A) Once the file (i.e.: Inventory.pdb) is created on your hard drive, simply use the HotSync Manager to install the file. Once you've told the HotSync manager to install the file, the file will be installed on your next HotSync. You can also read the JFile data that was on the device to perform your own syncing by reading it out of the Backups folder.

Q) Is there a way to programmatically install the file?

A) If you programmatically create the file in one of the following locations, it should automatically be installed on your next HotSync:

Mac OS X:

Macintosh HD:Users:gbailey:Documents:Palm:Users:Gerald Bailey:Files to Install:myJFileDatabase.pdb

Windows XP:

C:\Program Files\palmone:gbailey:Install:myJFileDatabase.pdb

Note: To programmatically perform HotSync operations, the developer simply needs to read the JFile database that was moved to the HotSync Backup Folder during the last HotSync from one of the following locations. It will contain all the changes last made on the handheld device. The developer can then use a combination of the Unique Record Number (saved in element zero of the data arrays), any primary key fields the developer saved with the record, and the Record Status (saved in the last element of the data arrays) to perform any necessary synchronization.

Mac OS X:

Macintosh HD:Users:gbailey:Documents:Palm:Users:Gerald Bailey:Files to Install:myJFileDatabase.pdb

Windows XP:

C:Program Files:palmone:gbailey:Backup:myJFileDatabase.pdb

4JFile Data Types

- 1 = String (also text)
- 2 = Boolean
- 4 = Date
- 32 = Time
- 8 = Long Integer
- 16 = Real
- 64 = Popup
- 70 = Calculated
- 73 = Date Popup - Creates Popup Date Picker
- 80 = Time Popup - Creates Popup Time Picker

4JFile Error Codes

- 0 Successful creation (or reading) of Palm database file.
- 1 Reading in the wrong PDB File Type
- 1 Too many records. JFile only supports up to 32,000 records.
- 2 The fieldsArray is not the same size as the dataPointersArray.
- 3 Filename parameter was not a text or string value.
- 4 The fildNamesArray is not the same size as the fieldTypesArray
- 5 The fildNamesArray is not the same size as the fieldAttributesArray
- 6 The size of the databaseOptionsArray is not 3
- 69 Unsupported Data Type

JFile Limitations

There are some limitations within JFile or the PalmOS itself. These are:
(JFile PDB Limitations - Ver. 5.6 Specs)

- Field names limited to 20 chars.
- 50 fields per database.
- 10,000 Characters per field.
- Maximum of 100 items per popup list.
- Maximum of 25 chars per item in popup list.
- Database names limited to 30 chars.

Have a question or comment?

Please see our website for more information regarding our products and services. We also provide a Frequently Asked Questions (FAQ) section which may answer your question.

<http://www.BaileyWare.com>

Contact us at:

Sales: sales@BaileyWare.com
Support: support@BaileyWare.com