

Apple II File Type Notes



Developer Technical Support

File Type: **\$19 (25)**
Auxiliary Type: **All**

Full Name: AppleWorks Data Base File
Short Name: AppleWorks DB File

Revised by: Matt Deatherage & John Kinder, CLARIS Corporation
Written by: Bob Lissner

July 1990
February 1984

Files of this type and auxiliary type contain an AppleWorks® Data Base file.

Changes since September 1989: Corrected the description of offset +337 in the header.

Files of type \$19 and any auxiliary type contain an AppleWorks Data Base file. AppleWorks is published by CLARIS. CLARIS also has additional information on AppleWorks files SEG.PR and SEG.ER. For information on AppleWorks, contact CLARIS at:

CLARIS Corporation
5201 Patrick Henry Drive
P.O. Box 58168
Santa Clara, CA 95052-8168

Technical Support
Telephone: (408) 727-9054
AppleLink: Claris.Tech

Customer Relations
Telephone: (408) 727-8227
AppleLink: Claris.CR

AppleWorks was created by Bob Lissner. AppleWorks 2.1 was done by Bob Lissner and John Kinder of CLARIS. AppleWorks 3.0 was done by Randy Brandt, Alan Bird and Rob Renstrom of Beagle Bros Software with John Kinder of CLARIS.

Definitions

The following definitions apply to AppleWorks files in addition to those defined for all Apple II file types:

MRL Data base multiple record layout

SRL	Data base single record layout
RAC	Review/Add/Change screen
DB	AppleWorks or /// E-Z Pieces Data Base
SS	AppleWorks or /// E-Z Pieces Spreadsheet
WP	AppleWorks or /// E-Z Pieces Word Processor
AW	AppleWorks or /// E-Z Pieces

Auxiliary Type Definitions

The volume or subdirectory auxiliary type word for this file type is defined to control uppercase and lowercase display of filenames. The highest bit of the least significant byte corresponds to the first character of the filename, the next highest bit of the least significant byte corresponds to the second character, etc., through the second bit of the most significant byte, which corresponds to the fifteenth character of the filename.

AppleWorks performs the following steps when it saves a file to disk:

1. Zeros all 16 bits of the auxiliary type word.
2. Examines the filename for lowercase letters. If one is found, it changes the corresponding bit in the auxiliary type word to 1 and changes the letter to uppercase.
3. Examines the filename for spaces. If one is found, it changes the corresponding bit in the auxiliary type word to 1 and changes the space to a period.

When files are read from disk, the filename and auxiliary type information from the directory file entry are used to determine which characters should be lowercase and which periods should be displayed as spaces. If you use the auxiliary type bytes for a different purpose, AppleWorks will still display the filenames, but the wrong letters are likely lowercase.

File Version Changes

Certain features present in AppleWorks 3.0 files are not backward-compatible to 2.1 and earlier versions. Such features are noted in the text. AppleWorks Data Base files which may not be loaded by versions prior to 3.0 are identified by a non-zero byte at location +218, referred to as location `DBMinVers`.

Those features added for AppleWorks 2.0, 2.1 and 3.0 not previously documented are indicated with that version number in the margin.

Data Base Files

Data base files start with a variable length header, followed by 600 bytes for each report format (if any), the standard values record, then variable length information for each record.

Header Record

The header contains category names, record selection rules, counts, screen positioning information, and all other non-record specific information.

	+000 to +001	Word	The number of bytes in the remainder of the header record. Use this count for your next ProDOS read from the disk.
	+002 to +029 +030	Byte	Ignore these bytes. Cursor direction when the Return key is pressed in SRL. \$01: Order in which you defined categories or \$02: Left to right, top to bottom.
	+031	Byte	What direction should the cursor go when you press the Return key in the MRL? D)own or R)ight.
	+032 to +033 +034	Byte	Ignore these bytes. Style of display that Review/Add/Change was using when the file was saved: R: SRL. Slash (/): MRL.
	+035	Byte	Number of categories per record. Values from \$01 to \$1E.
3.0	+036 to +037	Word	Number of records in file. If <code>DBMinVers</code> is non-zero, the high bit of this word may be set. If it is, there are more than eight reports and the remaining 15 bits contain the true number of records defined.
3.0	+038	Byte	Number of reports in a file, maximum of 8 (20 for 3.0).
	+039 to +041 +042 to +071	Bytes	Ignore these bytes. For each of up to 30 columns, showing the number of spaces used for this column on the MRL. Be sure you understand that categories may have been rearranged on the MRL. Byte +042 refers to the leftmost column on the MRL.
	+072 to +077 +078 to +107	Bytes	Ignore six bytes. For up to 30 categories on the MRL, the defined category that appears in each position. Byte +078 is the leftmost column of the MRL and has a value from \$01 to \$1E that defines which of the category names appears in this position. These numbers change as a result of changing the layout of the MRL.
	+108 to +113 +114 to +143	Bytes	Ignore six bytes. For up to 30 categories on the SRL, the horizontal screen position. These are changed as a result of changing the layout of the SRL. AppleWorks makes sure that these entries, and the vertical screen positions, are kept in order from left to right within top to bottom.
	+144 to +149 +150 to +179	Bytes	Ignore these six bytes. For up to 30 categories on the SRL, the vertical screen position.
	+180 to +185 +186 to +215	Bytes	Ignore six bytes. For up to 30 categories on the SRL, which of the category names appears in this position. These change as a result of changing the SRL. This number refers to the category names listed below.
	+216 to +217		Ignore two bytes.

3.0	+218	Byte	DBMinVers. The minimum version of AppleWorks needed to read this file. This will be \$00 unless there are more than 8 report formats; it will then contain the version number 30 (\$1E) or greater.
3.0	+219	Byte	The first frozen column in the titles.
3.0	+220	Byte	If this is zero, no titles are present. If non-zero, this is the last frozen column.
3.0	+221	Byte	Leftmost active column. This is zero-based; if this value is zero, it means column one, etc.
	+222	Byte	Number of categories on MRL. Will be less than or equal to the number of categories in the file. SRL displays all categories, so there is no equivalent number for SRL.
	+223 to +224	Word	For the first line of RAC selection rules. Zero means no selection rules, while any other value refers to the category name that is tested. The high byte will always be zero.
	+225 to +226	Word	Category name for the second line of RAC selection rules. Zero means that there is only one line.
	+227 to +228	Word	Category name for the third line of RAC selection rules. Zero means that there is no third line.
	+229 to +230	Word	For the first line of RAC rules, which of the tests is to be applied. 1 means equals, 2 means greater than and so on.
	+231 to +232	Word	Test for the second line of rules, if any.
	+233 to +234	Word	Test for the third line, if any.
	+235 to +236	Word	Continuation code for the first line: 1: And, 2: Or, 3: Through.
	+237 to +238	Word	Continuation code for the second line.
	+239 to +240	Word	Continuation code for the third line. Not possible, so it is always zero.
	+241 to +272	String	Maximum length of 30 bytes. Comparison information for the first line RAC selection rules.
	+273 to +304	String	Comparison for the second line.
	+305 to +336	String	Comparison for the third line.
	+337 to +356		Ignore these twenty bytes.
	+357 to +378	String	Name of the first category. Maximum length of 20 bytes. If the file has only one category, the header record will end here.
	+379 to +400	String	Name of the second category, if any. This area will not be on the header record if there is only one category.
	+401	22 Bytes	Additional 22 byte entries for all remaining categories. The size of the header record depends on the number of categories. Space is not maintained past the last category.

Report Records

Report records follow the header record. One of the header record categories tells you how many report records to expect. The number will be from zero to eight. Each report record is 600 bytes, and contains:

+000 to +019	String	Report name. Maximum length of 19 characters.
--------------	---------------	---

+020 to +052	Bytes	Column width for up to 33 columns in a tables-style report format. Byte +020 is for the leftmost column on a tables-style report. There can be up to 30 categories from the file, plus 3 more calculated columns. For labels-style report formats, the value is a byte that has the horizontal position of this category, relative to the left margin.
+053 to +055 +056 to +088	Bytes	Skip 3 bytes. For tables-style: Number of spaces to be printed at the right of justified columns. For labels-style: Vertical position on the report for each of up to 30 categories. A value of 1 means that category is on the first line of labels-style report. Skip 3 bytes.
+089 to +091 +092 to +124	Bytes	For up to 33 columns of tables-style: Values from 1 to 30 refer to which category name appears in this column on the report. Values of \$80, \$81 and \$82 are the three calculated categories, from left to right. For labels-style: Same as tables-style, minus the calculated categories. Skip these three bytes.
+125 to +127 +128 to +160	Bytes	For up to 33 columns of tables-style: \$99 means no foot totals, 0 through 4 means the number of decimal places for a foot total. For labels-style: For up to 30 categories on report, Boolean bytes whether or not category names are to be printed. Skip these three bytes.
+161 to +163 +164 to +196	Bytes	For up to 33 columns of tables-style: \$99 means left justified, 0 through 4 means right justified with 0 to 4 decimal places. For up to 30 categories of labels-style: Boolean bytes whether or not to float (OA-J) this category up against the category to its left. Skip three bytes.
+197 to +199 +200	Byte	Number of categories on report. Includes calculated categories, if any.
+201	Byte	Tables-style. If there is at least one calculated category, this contains values from 1 to 33: which column of the report. Labels-style: Values from 3 to 21. Position of the line on the screen that says "Each record will print nn lines."
+202	Byte	Tables-style: Same as +201, but for the second calculated category, if any. Labels-style: Unused.
+203	Byte	Tables-style: Same as +201, but for the third calculated category, if any. Labels-style: Unused
+204	Byte	Tables-style only: If there is a group total column, this byte states which of the category names is used as a basis. Values from 1 to 30.
+205	Byte	Platen width value, in 10ths of an inch. For example, a value of 8.0 inches entered by the user will show as 80 or \$50.
+206	Byte	Left margin value. All inches values are in 10ths.

+207	Byte	Right margin value.
+208	Byte	Characters per inch.
+209	Byte	Paper length value, in 10ths of an inch.
+210	Byte	Top margin value.
+211	Byte	Bottom margin value.
+212	Byte	Lines per inch. 6 or 8.
+213	Byte	Not relevant. Probably always a "C."
+214	Byte	Type of report format. H: tables-style, V: labels-style.
+215	Byte	Spacing: S(ingle, D(ouble, or T(riple. Expect these three letters, even in European versions.
+216	Byte	Print report header. Boolean.
+217	Byte	Tables-style: If user has specified group totals, Boolean, just print the group totals.
+218	Byte	Labels-style: Boolean, omit the line when all entries on the line are blank.
+219	Byte	Labels-style: Boolean, keep the number of lines the same within each record.
+220 to +301	String	80-byte string. Title line, if any.
+302 to +323	String	Tables-style. 20-byte string. Name of the first calculated category, if any.
+324 to +355	String	Tables-style. 30-byte string. Calculation rules for first calculated category, if any.
+356 to +409	String	Tables-style. Name and rules for second calculated category, if any.
+410 to +463	String	Tables-style. Name and rules for third calculated category, if any.
+464 to +477	String	If user has specified "Send special codes to printer," this is a 13 byte string containing those codes.
+478	Byte	Boolean: Print a dash when an entry is blank.
+479 to +592	Words & Strings	Record selection rules. Exact same format as described in the header record.
+593 to +599		Unused

Data Records

Data records follow the report records. The first data record contains the standard values. Each following data record corresponds to one data base record.

These records contain all of the categories within one stream of data. The category entries are in the same order that the category names appear in the header record.

Bytes +0 and +1 are a word that contains a count of the number of bytes in the remainder of the record.

Byte +2 of each record will always be a control byte. Other control bytes within each record define the contents of the record. Control bytes may be:

\$01-\$7F	This is a count of the number of following bytes that are the contents of a category.
\$81-\$9E	This (minus \$80) is a count of the number of categories to be skipped. For example, \$82 means skip two categories.
\$FF	This indicates the end of the record.

The information in individual categories may have some special coding so that date and time entries can be arranged (sorted).

Date entries have the following format:

+000	Byte	\$C0 (192). Identifies a date entry.
+001 to 002	Two bytes	ASCII year code, like "84" (\$38 \$34).
+003	Byte	ASCII month code. A means January, L means December.
+004 to +005	Two bytes	ASCII day of the month, like "31" (\$33 \$31).

Time entries have the following format:

+000	Byte	\$D4 (212). Identifies a time entry.
+001	Byte	ASCII hour code. A means 00 (the hour after midnight). X means 23, the hour before midnight.
+002 to +003	Two bytes	ASCII minute code. Values from 00 to 59.

File Tags

All AppleWorks files normally end with two bytes of \$FF; **tags** are anything after that. Although File Tags were primarily designed by Beagle Bros, they can be used by any application that needs to create or modify an AppleWorks 3.0 file.

Because versions of AppleWorks before 3.0 stop at the double \$FF, they simply ignore tags.

The File Tag structure is as follows:

+000	Byte	Tag ID. Should be \$FF.
+001	Byte	2nd ID byte. These values will be defined and arbitrated by Beagle Bros Software. Beagle may be reached at: Beagle Bros Inc 6215 Ferris Square, #100 San Diego, CA 92121
+002 to +003	Word	Data length. If this is the last tag on the file, the low byte (+002) will be a count of the tags in this file, and the high byte (+003) will be \$FF.
+004 to nnn	Bytes	Actual tag data, immediately followed by the next four-byte tag ID. These bytes do not exist for the last tag.

There is a maximum of 64 tags per file. Each tag may be no larger than 2K.

AppleWorks is a registered trademark of Apple Computer, Inc. licensed to Claris Corporation.