AppleSingle and AppleDouble format internals

AppleSingle format stores the data, resources, and attributes of a Macintosh file in a single A/UX file. AppleDouble format stores a file's data in one file and stores its resources and attributes in another file.

This section uses these terms:

- **Home file system** is the file system for which the file's contents were created, not necessarily the file system in which the file was created. The Macintosh file system is the home file system for all A/UX Toolbox applications and all documents created with A/UX Toolbox applications.
- **Foreign file system** is the other file system that will store or process the file. The UNIX file systems are the foreign file systems for all A/UX Toolbox applications and all documents created with A/UX Toolbox applications.

AppleSingle format

In AppleSingle format, all of a file's contents and attributes are stored in a single file in the foreign file system.

An AppleSingle file consists of a header followed by one or more data entries. The header consists of several fixed fields and a list of entry descriptors, each pointing to an entry. Table 6-3 describes the contents of an AppleSingle file header.

Table 6-3 AppleSingle file header

Field	Length
Magic number	4 bytes
Version number	4 bytes
Home file system	16 bytes, ASCII encoded
Number of entries	2 bytes
Entry descriptor for each entry:	
Entry ID	4 bytes
Offset	4 bytes
Length	4 bytes

Byte ordering in the file-header fields follows MC68000, MC68020, and MC68030 conventions. Here is a description of each field:

- *Magic number* This field, modeled after the A/UX magic-number feature, specifies the file's format. Apple has defined the magic number for AppleSingle format as 0x00051600.
- *Version number* This field allows for the evolution of AppleSingle format. This section describes version 0x00010000.
- *Home file system* This field defines the home file system. It contains a 16-byte ASCII string, which is not preceded by a length byte but which can be padded with spaces. Apple Computer has defined these strings:

Macintosh	`'Macintosh'`	or	`0x4D616369 0x6E746F73 0x68202020 ...`
ProDOS	`'ProDOS'`	or	`0x50726F44 0x4F532020 0x20202020 ...`
MS-DOS	`'MS-DOS'`	or	`0x4D532D44 0x4F532020 0x20202020 ...`
UNIX	`'Unix'`	or	`0x556E6978 0x20202020 0x20202020 ...`
VMS™	`'VAX VMS'`	or	`0x56415820 0x564D5320 0x20202020 ...`

  All A/UX Toolbox applications work with files whose home file system is Macintosh.
- *Number of entries* This field reports how many different entries are included in the file. Its value is an unsigned 16-bit number. If the number of entries is any number other than 0, then that number of entry descriptors immediately follows.
- *Entry ID* This field defines what the entry is. The field holds an unsigned, 32-bit number. Apple Computer has defined a set of entry IDs and their values:

Data fork	1	standard Macintosh data fork
Resource fork	2	standard Macintosh resource fork
Real name	3	file's name in its home file system
Comment	4	standard Macintosh comments
Icon, B&W	5	standard Macintosh black-and-white icon
Icon, color	6	Macintosh color icon
file info	7	file information: attributes and so on
Finder info	9	standard Macintosh Finder information

  Apple reserves the range of entry IDs from 0 to 0x7FFFFFFF. The rest of the range is available for other definitions. Apple does not arbitrate the use of the rest of the range.

Icon entries do not appear in most files because they are typically stored as a bundle in the resource fork of the application file.

The structure of the "file info" entry is different for each home file system. For Macintosh HFS files, the entry is 16 bytes long and consists of three long-integer dates (create date, last modification date, and last backup date) and a long integer containing 32 Boolean flags. Where 0 is the least-significant bit and 31 is the most-significant bit, bit 0 of the Macintosh "file info" entry is the Locked bit, and bit 1 is the Protected bit. Figure 6-5 illustrates the formats for Macintosh HFS, A/UX, MS-DOS, and ProDOS "file info" entries.

The "Finder info" entry consists of 16 bytes of Finder information followed by 16 bytes of extended Finder information (the fields `ioFlFndrInfo` followed by `ioFlXFndrInfo`, as returned by the `PBGetCatinfo` call). These fields contain extended-file-attribute information. See *Inside Macintosh,* Volume VI, for a description of the subfields in these fields. Newly created files contain zeros in all "Finder info" fields. When you are creating a file whose home file system is Macintosh, you can use 0 in any subfield whose value is unknown, except that you should set the `fdType` and `fdCreator` subfields. Values should be set by means of standard File Manager calls such as `SetFInfo` and `PBSetCatInfo`.

- *Offset* This field contains an unsigned 32-bit number that shows the offset of the beginning of the entry's data from the beginning of the file.
- *Length* This field contains an unsigned 32-bit number that shows the length of the data in bytes. The length can be 0.

The entry data follows all of the entry descriptors. The data in each entry must be in a single, contiguous block. You can leave holes in the file for later expansion of data. For example, even if a file's comment field is only 10 bytes long, you can place the offset of the next field 200 bytes beyond the offset of the comment field, to leave room for the comment to grow to its maximum length of 200 bytes.

The entries can appear in any order, but you can maximize the efficiency of file access by following these recommendations:

- Put the data fork entry at the end of the file. The data fork is the most commonly extended entry, and it is easier to increase its length if it is the last thing in the file.
- Put the entries that are most often read, such as "Finder info," as close as possible to the header, to increase the probability that a read of the first block or two will retrieve these entries.

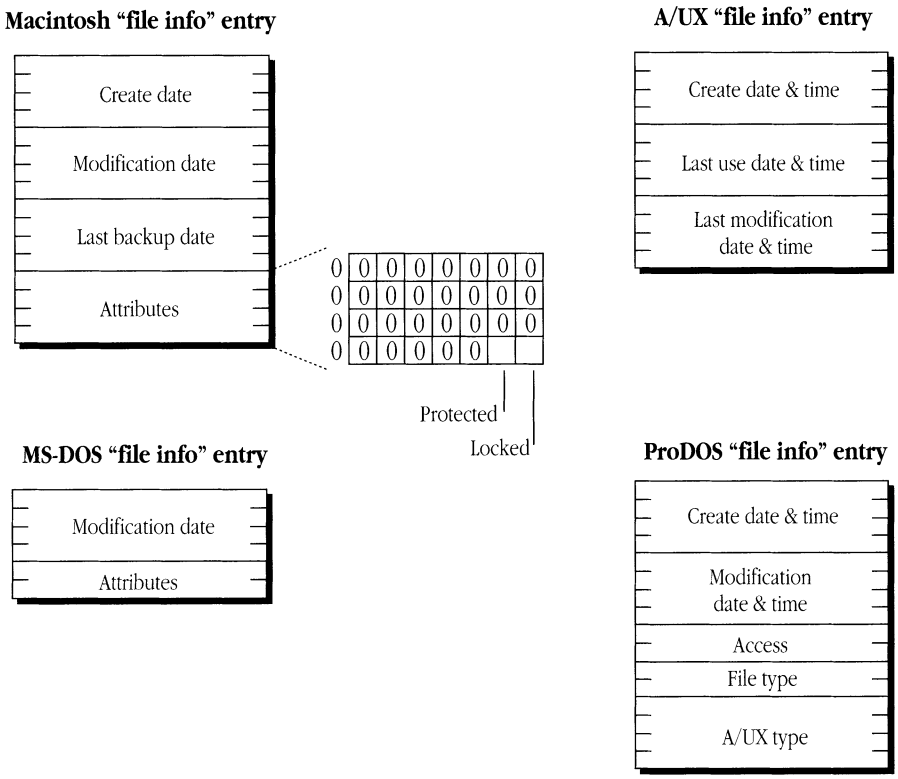

Figure 6-5 Formats for "file info" field entries

AppleDouble format

In AppleDouble format, the file's data fork is stored in a file called the *AppleDouble data file,* and the file's attributes and resources are stored in a separate file called the *AppleDouble header file.*

The AppleDouble data file contains the data fork, in exactly the form in which it appears in a Macintosh file, with no extra header.

The AppleDouble header file has the same format as an AppleSingle file, except that it contains no data fork entry. The magic number for an AppleDouble header file is 0x00051607. The entries in the header file can appear in any order. It is usually more efficient to put the resource fork at the end of the file because the resource fork is the entry most likely to expand.