Foreign File Formats: AppleSingle and AppleDouble (9/94)

Article Created: 29 April 1988
Article Reviewed/Updated: 21 September 1994

TOPIC ---

Apple provides two standards for representing files on foreign file systems.
The goal is to preserve all attributes of the file's home file system on file
systems that don't otherwise support the same attributes.

DISCUSSION --

Experience indicates that one format is inadequate to cover all cases. Two
closely related formats, however, can serve most needs. The primary reason for
developing these formats is to allow storage of Macintosh files on file systems
that don't support the notion of two forks. However, the proposed formats are
general enough for use in representing a file from any file system on any other
file system.

AppleSingle

AppleSingle format keeps all contents and attributes of a file in a single file
on the foreign file system. For example, both forks of a Macintosh file, the
Finder information and associated icons, and so on, are arranged in a single
file with a simple structure. The intention is to use this format primarily as
a storage format; that is, for Macintosh files that must be stored on a foreign
file system and later reconstructed into true Macintosh files.

An AppleSingle file contains a header followed by data. The header consists of
several fixed fields and a list of "entry descriptors," each pointing to an
"entry." Apple defines these standard entries as Data Fork, Resource Fork, Real
Name (name in the home file system), Comment, Icon, and File Info. Each entry
is optional and may or may not appear in the file.

AppleDouble

AppleDouble format is more appropriate for applications in which the users of
the foreign file system want to modify the contents of the file. Since most
Macintosh applications keep the file data in the data fork, AppleDouble format
saves the contents of the data fork in one file, and all other file attributes
are kept in a separate file.

AppleDouble format is the same as AppleSingle format, except that the data is kept in a separate foreign file. The file containing the data fork is called the "AppleDouble Data File," and the other file is called "AppleDouble Header File."

The AppleDouble Data file contains only the standard Macintosh data fork, with no extra header. The AppleDouble Header File has exactly the same format as the AppleSingle file, except that it DOESN'T contain a data fork entry.

Applications

• Standard for Transferring Files Between Host Computers (Interhost)
  For example, Macintosh files could be shipped easily and completely among
  different systems if they all understood one of these common formats.
  Any existing e-mail system or file transfer utility could be used without
  modification.

• Standard for Operating on Foreign Files within a Single Host (Intrahost)
  A possible example would be UNIX applications that can build and
  manipulate Macintosh resource forks (perhaps a cross-development system).
  If host computer users want to write Macintosh-aware applications, they
  can use a common storage format, such as AppleSingle and AppleDouble.

AppleSingle and AppleDouble Specifications

There is a document titled "AppleSingle/AppleDouble Formats for Foreign Files
Developer's Notes." It describes version 2 of the AppleSingle/AppleDouble file
formats. These formats were originally developed for storing Macintosh files on
an A/UX file system, but have since been adapted for storing files from various
operating systems (not just Macintosh) on foreign file systems.

You can obtain a printed version of this document from APDA. You can also find
this document in the Technical Information Library under the title
"AppleSingle/AppleDouble Formats: Developer's Note".

Article Change History:
21 Sep 1994 - Removed binary file.
02 Feb 1993 - Updated.
06 Jan 1993 - Revised to move 'fcnvt' information to a separate article.

Support Information Services
Copyright 1988-94 Apple Computer, Inc.

Keywords: <None>

==

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 2894