Technical Note FL36
Apple Extensions to ISO 9660

CONTENTS

This Technical Note describes extensions Apple has defined to ISO 9660 in order to better support Macintosh file interchange. ISO 9660 is an international standard for formatting CD-ROM discs. This tech note is only important if you are a publisher of authoring tools for ISO 9660 discs, or are interested in details of how the Macintosh supports ISO 9660.

[Jul 01 1989]

Introduction

It may be desirable to create an ISO 9660 CD ROM containing Macintosh files in order to benefit by the storage capacity and distribution cost savings of CD ROM, and the interchange advantages of ISO 9660. However, the Macintosh file system requires information that the ISO 9660 file system does not support: HFS requires a file type, a file creator, and file attributes.

This document defines a protocol which extends the ISO 9660 specification to include HFS specific information, without corrupting the ISO 9660 structures. Discs created using the protocol are valid ISO 9660 discs, and should not behave differently on non-Apple receiving systems.

The protocol was designed to solve existing compatibility problems as well as allow for future expansion. It uses the `SystemIdentifier` field in the Primary Volume Descriptor for global information, and the `SystemUse` field in the directory record for file specific information.

Back to top

ISO 9660 compared to HFS

The Apple Macintosh file system is named "HFS", for "Hierarchical File System." This file system is similar to both MS-DOS and Unix in supporting subdirectories, where files may be logically grouped together. Subdirectories on the Macintosh are called folders.

When the Macintosh was first introduced, a flat file system called "MFS" for "Macintosh File System" was used. The MFS file system did not adequately provide for high capacity storage media such as hard disks and CDROM, so the HFS format was introduced in its place. MFS format is now generally used only on 400K, single sided floppy disks. HFS is compatible with MFS.

Macintosh file and volume names are different from both Unix and MS-DOS in that they allow any character except the

colon ":" as a valid file name character. This means that a name such as "My payroll file" is perfectly acceptable on the Macintosh. There is no concept of a file name extension, such as is found in the MS-DOS or ISO 9660 formats. HFS does not distinguish between upper and lower case names; the names "forecast", "Forecast", and "FoReCaSt" all refer to the same file. Volume names may have a maximum of 27 characters, and file names may have a maximum of 31 characters. To specify a file name in a folder (subdirectory), you give the folder name, followed by a colon, followed by the file name. For example, if you have the file "My payroll file" in the folder "business", you could specify the file as "business:My payroll file". To specify a file in the folder immediately above the current folder, you may use the shortcut name of two colons, "::".

An area where HFS is different from MS-DOS or Unix is in the concept of file forks. A file has two forks; a resource fork and a data fork. The data fork is used for the same purpose as a "file" in MS-DOS or Unix; it contains information in an unstructured format. The data fork is used by an application to store the contents of the document.

The resource fork of a file contains Macintosh resources. Resources are data in a special format; Apple Computer has defined many "famous" types of resources, and we have provided facilities so that programmers may define their own custom resource types. The resource fork of a file is accessed using special program calls, from a part of the Macintosh operating system known as the Resource Manager. All resources are labeled with a resource type of four characters. Some examples of resources are: 'CODE', for executable code; 'ICON' for a 16 bit by 16 bit icon; 'ICN#', a slightly different icon resource used by the Finder to display an unique icon for a file; 'MENU' for the menus found at the top of every Macintosh application; and 'STR ', for strings used within a program. There are many other types of resources as well.

Resources provide the Macintosh with a flexibility not found in other operating systems. For instance, to "localize" a well-written application into another language, all that needs to be changed are the resources that are language specific (usually the 'MENU' and 'STR ' resources.) Apple provides developers with several tools and resource editors that assist in modifying resources.

As well as the resource and data forks, additional information is kept about each file. This information is used by the Finder to display the position, attributes, and icon of a file. The additional information consists of a file type of four characters (such as 'TEXT' for plain text or unknown type documents, 'APPL' for applications, 'MACA' for MacWrite documents, and so forth), a file creator unique to each application (such as 'MWRT', 'EXCL', 'FNDR') and file attributes (such as invisible, system file, or locked against accidental deletion.) The additional information also includes a mapping of file creator to appropriate application, so that a Macintosh user may select a document and open it, and the appropriate application will be invoked and passed the document name.

Back to top

Problems with ISO 9660 Support on the Macintosh

Apple supports the ISO 9660 format through the use of a feature in the Macintosh file system called the "external file system" hook. This is a special low-memory global which contains a pointer to a list of external file system handlers. For ISO 9660 support, Apple has written a new set of routines contained in a file called "Foreign File Access". This file, plus the files "High Sierra File Access" and "ISO 9660 File Access" combine to provide complete support for the ISO 9660 format and its predecessor, High Sierra. Apple's ISO 9660 software supports level 2 interchange, according to section 10.2 of the ISO 9660 specification. This means that Apple does not yet support interleaved files or multi-disc volumes.

The Apple extensions do not fully correct deficiencies in the ISO 9660 specification; some issues are handled by the ISO 9660 File Access software. For instance, in HFS, the Finder keeps track of the position of a file on the desktop or in a folder by using a special field inside HFS; under ISO 9660, a file's position is computed when the folder is opened, and can not be changed.

Because of some deficiencies in the original design of the Finder, it is not possible to have the correct icons display for a file on a ISO 9660 disc. This is because the Finder does not actually ask for the icon of a file; rather, it assumes the existence of a *desktop database* which contains these mappings, and makes a special call, giving only the file creator and type. The software to provide this information was designed to be very HFS specific. Apple Computer will address this defect in a future release of the operating system. Currently, even if a file's icon bitmap is defined in the Apple extensions, it will not be used by the Finder. All files that reside on a ISO 9660 disc will display a generic icon. If such a file is copied to a hard disk, the correct icon will be displayed. You can still double-click on a ISO 9660 file, and it will open correctly.

If a ISO 9660 disc has been pressed without the Apple extensions, all files on the disc are considered to be of type 'TEXT' and creator 'hscd'. Type 'TEXT' is a generic type which can be read successfully by many Macintosh applications.

The Directory Record SystemUse Field

Directory records in the ISO 9660 specification have the following format. See section 9.1 of ISO 9660 for specific details.

```
byte    DirectoryRcdLength
byte    XARlength
 struct    ExtentLocation
struct    DataLength
struct    RecordingDateTime
byte    FileFlags
byte    FileUnitSize
byte    InterleaveGapSize
long    VolumeSequenceNum
byte    FileNameLength
char    FileName[FileNameLength]
byte    RecordPad (if necessary)
char    SystemUse[SystemUseLength]
byte    RecordPad (if necessary)
```

The RecordPad field is present only if needed to make DirectoryRcdLength an even number. If present, the RecordPad field *must* be zero ($00). The SystemUse field is an optional field under ISO 9660; it is defined for our use below.

The SystemUse field, when present, must begin with a signature word , followed by a one byte length and a one byte SystemUseID, followed by file system specific information. This structure may be repeated multiple times for multiple file systems; however, the total length of the SystemUse area (SystemUseLength) *must* be an even number. The signature word allows a receiving system to ensure that it can interpret the following data correctly, and the SystemUseID determines the type and format of the information which follows.

CD-ROM XA discs must be handled as a special-case in that the signature word, if present, will begin at byte 14 of the SystemUse field rather than at byte 0.

There are two AppleSignatures. The preferred AppleSignature is defined as "AA" ($41 41). The old AppleSignature, used for a previous version of this document, was defined as "BA" ($42 41). The old AppleSignature is the only one supported by the Macintosh AppleCD SC software version 2.0. Disks pressed using the old format are still supported, but we recommend that new disks be pressed using only the current format.

Receiving systems must perform a simple calculation to determine if the SystemUse field is present in any given directory record. It is present if:

```
DirectoryRcdLength - FileNameLength > 34
```

Receiving systems should first verify that the SystemUse field is present (making sure to account for the possibility of a RecordPad field), then check for the AppleSignature before interpreting the SystemUseID.

The SystemUseID is defined as follows for an AppleSignature of "AA". This is the version of the Apple Extensions that is supported by future versions of the Macintosh AppleCD SC software.

SystemUseID Definition

$00 reserved.

$01 ProDOS file_type and aux_type follow.

$02 HFS fileType, fileCreator, finder flags

$03-FF reserved.

The following tables define in detail the data formats for each SystemUseID. The MSB-LSB notation ("MSB" = Most Significant Byte, "LSB" = Least Significant Byte) means that the MSB occupies the lowest address, while LSB-MSB means that the LSB occupies the lowest address.

SystemUseID 01, ProDOS: SystemUse offset Contents

$00-01 $41 41 (AppleSignature)

$02 SystemUse Extension length ($07 for this ID, includes AppleSignature bytes)

$03 $01 (SystemUseID)

$04 ProDOS file type

$05-06 ProDOS aux type (LSB-MSB)

Note that a padding byte must be added if this ID is the only extension added to the directory record.

SystemUseID 02, HFS: SystemUse offset Contents

$00-$01 $41 $41 (AppleSignature)

$02 SystemUse Extension length ($0E for this ID, includes AppleSignature bytes)

$03 $02 (SystemUseID)

$04-$07 HFS fileType (MSB-LSB)

$08-$0B HFS fileCreator (MSB-LSB)

$0C-$0D HFS finder flags (MSB-LSB)

The SystemUseID is defined as follows for an AppleSignature of "AA". This is the version of the Apple Extensions that is supported by the Macintosh AppleCD SC software version 2.0.

SystemUseID Definition

$00 reserved.

$01 ProDOS file_type and aux_type follow.

$02-05 obsolete versions of HFS; do not use

$06 HFS fileType, fileCreator, finder flags

$07-FF reserved.

The following tables define in detail the data formats for each SystemUseID. The MSB-LSB notation ("MSB" = Most Significant Byte, "LSB" = Least Significant Byte) means that the MSB occupies the lowest address, while LSB-MSB means that the LSB occupies the lowest address.

SystemUseID 01, ProDOS: SystemUse offset Contents

$00-01 $42 41 (AppleSignature)

$02 $01 (SystemUseID)

$03 ProDOS file type

$04-05 ProDOS aux type (LSB-MSB)

SystemUseID 06, HFS: SystemUse offset Contents

$00-$01 $42 $41 (AppleSignature)

$02 $06 (SystemUseID)

$03-$06 HFS fileType (MSB-LSB)

$07-$0A HFS fileCreator (MSB-LSB)

$0B-$0C HFS finder flags (MSB-LSB)

The authoring software can simply copy the finder flags as retrieved by the HFS call PBGetFInfo. Only bits 5 (always switchLaunch), 12 (system file), 13 (bundle bit), and 15 (locked) are used. All other bits are either ignored or set due to internal workings of the file system translator. See *Inside Macintosh:Files* for more details about the finder flags.

There has been some confusion about padding bytes in a directory record which also contains Apple extensions. There are two possible places in a directory record where a padding field may occur: after the File Identifier, and at the end of the directory record. If no Apple extensions exist for a file, then the two locations are identical, and the rule stated in section 9.1.12 applies: "...(a padding field) shall be present ...only if the number in the Length of the File Identifier is an even

number." If Apple extensions exist, then section 9.1.13 *also* applies: "If necessary, so that the Directory Record comprises an even number of bytes, a (00) byte shall be added to terminate this field." (i.e. the System Use field used for Apple extensions.)

Back to top

The Extension to ISO 9660

This section describes the extension to ISO 9660 which allows multiple users of the "System Use" field. All references are to ISO 9660, First edition, 1988-04-15.

Section 9.1.13 System Use [PB (LEN_DR - LEN_SU + 1) to LEN_DR] shall be replaced as follows:

This field shall be optional. If present, this field shall be reserved for system use. If necessary, so that the Directory Record comprises an even number of bytes, a (00) byte shall be added to terminate this field.

If this field is present, it shall be broken up into a series of System Use Extensions. There can be more than one System Use Extension for a given directory record. A System Use Extension shall have the following format:

BP 1 : byte 1 of the signature. This field shall contain an 8-bit number. This field shall be recorded according to 7.1.1.

BP 2 : byte 2 of the signature. This field shall contain an 8-bit number. This field shall be recorded according to 7.1.1.

BP 3 (LEN_SE): This field shall specify as an 8-bit number the length in bytes of this System Use Extension. This field shall contain an 8-bit number. This field shall be recorded according to 7.1.1. This field shall include the length of the two signature bytes preceeding this byte.

BP 4 to LEN_SE : this field shall be reserved for system use. Its content is not specified by this International Standard.

Multiple System Use Extension fields may exist for a given directory record, subject only to the limitation that the total length of a directory record must be able to be recorded in the 8-bit field defined in section 9.1.1.

Note: For CD-ROM XA discs, the Byte Positions listed above must be adjusted upwards by 14 to account for XA's 14 fixed length system use field.

Back to top

Guidelines For Transforming HFS Filenames

Since most HFS filenames are illegal in the ISO 9660 specification, some transformation will be required when creating an ISO 9660 disc with HFS files. No reversible transformation is possible without degrading performance; therefore, we can only define guidelines for publishers and authoring tool publishers to follow when performing the transformation:

- convert all lowercase characters to uppercase.

- replace all illegal characters, including periods, with underscore ("_", $5F).

- if the filename must be shortened, truncate the rightmost characters.

- if the filename refers to a file, append the characters ".;1" ($2E, 3B, 31).

Following these guidelines will result in more consistent discs.

ISO 9660 Associated Files

Associated files are exactly analogous to resource forks. Though the format of associated files is clear in the ISO 9660 specification, we would like to re-state it here:

An associated file is defined as having the associated bit set in the file flags byte of the directory record. It has exactly the same file identifier as its counterpart, and resides *immediately before* its counterpart in the directory. The associated file is treated as the resource fork, its counterpart is treated as the data fork of the file.

For example, if a file "FOO.;1" has an associated file, there will be two adjacent directory records named "FOO.;1"; the first one (the resource fork) will have the associated bit set, the second one (the data fork) will have the associated bit clear.

Back to top

Version numbers in ISO 9660

Section 13.3.2 of ISO 9660 says, "The implementation [of a receiving system, i.e. one using the CD] shall make available to the user the information that is recorded in each of the descriptor fields listed below." It then follows with a list for various records. For a directory record, the only things listed are file name, file extension, and a bit indicating whether the "file" is a directory. So that's the minimum that an ISO 9660 compliant translator must provide: a file name of the form XXX.YYY with everything specified as uppercase alphanumerics.

Section 12.3.1 of ISO 9660 says that a file descriptor includes the file name, file extension, and version number. This means that file descriptors look like FORECAST.DATAFILE;1 (or, for level 1 compliance, FORECAST.DAT;1) Apple's ISO 9660 support included the ";1" as part of the name of the file, because the specification appeared ambiguous, and because the name was valid under the Macintosh OS with the semicolon and version number attached.

Subsequent history has shown that this leads to problems with interchanging files with non-Macintosh systems. Starting with CD-ROM software release 5.0, the version number is not displayed unless you hold down the option key while mounting the CD-ROM disc.

Back to top

Sample Code for Retrieving Macintosh File Information

This code is an example of how to access the Macintosh file type, file creator, and finder flag information. The code is written in the C language.

```
/*****************************************************************************
 *
 *  Function:        GetFileInfo
 *
 *  Purpose:        get lengths of file rsrc and data forks, type, creator,
 *            and finder flags
 *
 *  Returns:        OSErr
 *            noErr, unless PBGetFInfo has an error.
 *
 *  Side Effects:    rsrcLength, dataLength, fType, fCreator, flags are
 *            updated with correct values for the file requested
 *
 *  Description:    call PBGetFInfo() and return its results.  This routine
 *            will only work if the path name to the file on the Mac
 *            can fit in 255 characters (the length of a pascal
 *            string)  See Inside Macintosh, Files for more information.
 *
 *****************************************************************************/
OSErr
GetFileInfo(name, vRefNum, rsrcLength, dataLength, fType, fCreator, flags)
StringPtr    name;
short        vRefNum;
long        *rsrcLength;
long        *dataLength;
OSType       *fType;
OSType       *fCreator;
short        *flags;
{
    HParamBlockRec    io;
    OSErr            result;

    io.fileParam.ioCompletion = 0L;
    io.fileParam.ioNamePtr = name;
    io.fileParam.ioVRefNum = vRefNum;
    io.fileParam.ioFVersNum = 0;
    io.fileParam.ioFDirIndex = 0;
    result = PBGetFInfo(&io, false);
    if (result == noErr)
    {
        *rsrcLength = io.fileParam.ioFlRLgLen;
```

```
        *dataLength = io.fileParam.ioFlLgLen;
        *fType = io.fileParam.ioFlFndrInfo.fdType;
        *fCreator = io.fileParam.ioFlFndrInfo.fdCreator;
        *flags = io.fileParam.ioFlFndrInfo.fdFlags;
    }
    else
    {
        *rsrcLength = 0L;
        *dataLength = 0L;
        *fType = 0L;
        *fCreator = 0L;
        *flags = 0;
    }
    return result;
}
```

Back to top

References

Apple CD-ROM Handbook

DV 22 -- CD-ROM Driver Calls

Apple IIgs GS/OS Reference from Addison-Wesley (for Apple II support of ISO 9660.)

Back to top

Change History

07-July-1988	Initial release.
14-July-1988	Removed reference to Macintosh in footnote. Modified note in ProDOS Filename Transformation section to require that Volume Identifier betransformed.
22-July-1988	Changed Protocol Identifier to all uppercase.
2-August-1988	Added SystemUseID 06 for Macintosh Finder flags.
13-August-1988	Corrected location of RecordPad field so comes before the SystemUse field. Added padding fields to SystemUseID's 2-5.
6-March-1989	To accommodate other receiving systems using the SystemUse field, the identification has been slightly modified. A new signature has been added, with some unused fields in the Macintosh extensions removed.
6-April-1989	Described when the Protocol Identifier is required and how to use the extensions on CD-ROM XA discs.
30-May-1989	Corrected SystemUse format description for HFS (SystemUseID 02).
21-July-1989	Added warning about Macintosh versions and what they support. Put back in description of old AppleSignature for reference.
8-February-1994	Converted to a Technical Note. Previously, this document was "hidden" on the developer CD in a technical report called "CD-ROM and the Macintosh Computer." Removed most ProDOS specific information; it is documented in the *Apple IIgs GS/OS Reference* from Addison-Wesley.

Back to top

Downloadables

Acrobat version of this Note (K) Download

Back to top

Technical Notes by Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code