WARNING!   THIS DESCRIPTION IS INCOMPLETE, AND DOES NOT DESCRIBE THE FINAL FORM OF THE CLASS

PICTFile

INHERITS FROM	File
DECLARED IN	PICTFile.h

CLASS DESCRIPTION

The PICTFile class is a simple subclass of the File class.   It provides an intermediate way to access Macintosh PICT files (read-only, at present).   Specifically, it allows one to read in PICT data structures (opcodes, rects, points, region components, etc) atomically.   I say `intermediate', because a more `advanced' pict file would doubtlessly read in opcodes and all their arguments and return them in specialized objects to the caller.   That behavior is beyond the scope of this class, however.

When the class is issued an order to open a file, it does three things immediately: First, it caches the 512byte header of the pict file, it then stores its bounding rect, and finally it records the version number of the PICT file.   Note that it ignores the file size (as this doesn't seem to be a 100% reliable number anymore).   It does all this, for this class assumes that it is working with a read-only stream of bytes that it can read only once.   This assumption is being made because I know this situation will occurr when the RTF converter program gets written [turns out that this won't be the case with the initial RTF converter after all].   It is, however, not a very nice assumption to make in general, and yet another indication that after all the converters get written, I should take what I've learned and redesign some aspects of these things.   in any case, at this point, the object leaves the current file position just after the version opcode and its arguments, and returns.

The caller, at this point, is free to read the data as suits their whim.   The PICTFile will, however, cease returning anything except errors after it has read and returned the FF

terminating opcode.   Therebye, hopefully keeping some programs, running amuck, from getting too carried away.

Aside from these features, the PICTFile provides all the features (including the brain-dead 'is the file there' method) that the File class supports.

INSTANCE VARIABLES

Inherited from File   <<various   (this is not a 'final' class)>>

Declared in PICTFile   BinaryString   PICTHeader;
                                             MacRect   boundingRect;
                                             Integer   PICTVersion;
                                             Boolean   foundEnd;

PICTHeader   The 512 byte header to the pict file

boundingRect   The rectangle bounding the picture

PICTVersion   The version of the pict file

foundEnd   Set to YES when the [00]FF opcode is found

METHOD TYPES

Opening the file   - initAndUse
 - initAndUseTemporary
 - OpenWithAccess

Retrieving info about file   - GetVersion
                                             - GetHeader
                                             - GetBounds

Reading PICT data types   - GetOpcode
                                             - GetRect
                                             - GetPoint
                                             - GetByte
                                             - GetPString
                                             - ????

CLASS METHODS

None

INSTANCE METHODS

initAndUse
 - (id) **initAndUse:** (ConstCString) *filename*

This is a simple subclass-ed method from the File class.   We subclass it because we need to initalize our instance variables.

initAndUseTemporary
 - (id) **initAndUseTemporary**

This is a simple subclass-ed method from the File class.   This method essentially implies doing write access to a file, which we don't presently support, so we just return an error.

OpenWithAccess:
 - (id) **OpenWithAccess:** (AccessType) *operation*

This is a simple subclass-ed method from the File class.   This simply traps to make sure that the *operation* is not one that involves file writing, as this is not supported at present by the PICTFile class.   If it is an operation involving writing, and error is set, and this returns self.

GetVersion
 - (Integer) **GetVersion**

Returns the version number of the PICT file to the caller.   This will be the number 1 for a version 1 pict file, 2 for a version two one, 0 for unknown, and 9999 for anything else (read: errors).

GetHeader
 - (ConstByteString) **GetHeader**

Returns a pointer to the 512 byte header of the PICT file.   The 512 bytes are not terminated in

any way, and it is assumed the caller is aware of this.

GetBounds
- (PICTRect) **GetBounds**

Returns a rectangle struct to the caller containing the boundrary rectangle of the PICT image in this PICT file.   The caller is responsible for disposing of the rectangle.

GetOpcode:
- (PICTopcode) **GetOpcode**

Reads the next opcode from the PICT file and returns it.   This reads either a one or two byte opcode, depending on whether it's version 1 or 2 (or above).   If the version is 2 or above, it will also automatically move to the next even word boundrary to read the next opcode.

GetRect
- (MacRect) **GetRect**

Reads an 8 byte PICT rectangle from the PICT file, and returns this copy to the caller.

GetPoint
- (PICTRect) **GetPoint**

Reads a 4 byte point from the PICT file and returns a copy to the caller.

GetByte
- (Byte) **GetByte**

This is just like ReadByte in the File class.   It reads a byte and returns it to the caller.   This is provided for naming consistancy with the other methods here.

GetPSstring
- (CString) **GetPString**

This reads in a Pascal style string from the PICT file, and returns the copy as a CString structure to the caller.

BUGS AND PROBLEMS

As noted above, the read-only aspect of this class is not ideal.   Moreover, the awkward method of dealing with it potentially being a one-pass stream is very icky.   Need a more general way for dealing with this transparantly to the class.
We are not subclassing the various Create... methods, and thus do not prevent someone from nuking a file we were only supposed to be able to read..

ENHANCEMENT IDEAS

See above.

CONSTANTS AND DEFINED TYPES

MODIFICATION HISTORY

$Log: PICTFile.rtf,v $Revision 1.7   93/04/04   23:29:49   deathSun Apr   4 23:29:49 PDT 1993Revision 1.6   93/01/09   21:07:06   deathSat Jan   9 21:07:06 PST 1993Revision 1.5   93/01/01   11:51:14   deathFri Jan   1 11:51:14 PST 1993Revision 1.4   92/12/31   15:34:03   deathThu Dec 31 15:34:03 PST 1992Revision 1.3   92/12/05   23:06:59   deathSat Dec   5 23:06:59 PST 1992Revision 1.2   92/12/03   18:01:18   deathThu Dec   3 18:01:17 PST 1992Revision 1.1   92/11/27   19:37:29   deathFri Nov 27 19:37:28 PST 1992