DirInfo structure

#include <Files.h>

typedef struct **DirInfo** {		Size	Offset	Description
ParamBlockHeader		24	0	common fields of ParamBlock types
short	ioFRefNum;	2	24	File reference number
short	filler1;	1	27	(unused)
short	ioFDirIndex;	2	28	Index
char	ioFlAttrib;	1	30	File Attribute bits (locked, directory, etc)
char	filler2;	1	31	(unused)
DInfo	ioDrUsrWds;	16	32	(Folder rectangle, location, flags, etc.)
long	ioDrDirID;	4	48	'Hard' directory ID
unsigned short	ioDrNmFls;	2	52	Number of files and directories in this dir
short	filler3[9];	18	54	(unused)
unsigned long	ioDrCrDat;	4	72	Date/Time of creation
unsigned long	ioDrMdDat;	2	76	Date/Time of last modification
unsigned long	ioDrBkDat;	4	80	Date/Time last backed up
DXInfo	ioDrFndrInfo;	16	84	(Scroll point, put-away dir, comment, etc.)
long	ioDrParID;	4	100	'Hard' ID of this dir's parent
} **DirInfo**;		104		

Notes: Use this DirInfo structure in calls to **PBGetCatInfo** and **PBSetCatInfo** when you access information about a directory.

A common way to use this structure is to allocate a CInfoPBRec union which is an aggregate of HFileInfo and DirInfo. Create and initialize a pointer to each data type and use either structure in the call to **PBGetCatInfo**. Upon return, check bit 4 of ioFlAttrib. If bit 4 is set, then the return data is about a directory and you should use the DirInfo structure; otherwise, use HFileInfo. See CInfoPBRec for examples.