

PBDTGetIconInfoRetrieve an icon type and associated file type

#include <Files.h>

Finder Interface

OSErr **PBDTGetIconInfo**(*paramBlock*);
DTPBPtr *paramBlock* ; pointer to a DTPB Param Block
Boolean *async*; 0 = await completion; 1 = immediate return

Parameter block

→	12	ioCompletion	long	completion routine
←	16	ioResult	short	result code
→	24	ioDTRefNum	short	database reference number
→	26	ioIndex	short	index into icon list
←	28	ioTagInfo	long	reserved; must be initialized to 0
←	40	ioDTActCount	long	size of icon bitmap
←	45	ioIconType	char	icon type
→	52	ioFileCreator	long	icon's <u>file creator</u>
←	56	ioFileType	long	icon's <u>file type</u>

PBDTGetIconInfo retrieves the icon type and the associated file type of an icon in the database. You use it to identify the set of icons associated with a given creator. You specify the creator by placing its signature in ioFileCreator, and you specify the database by placing the desktop database reference number in the ioDTRefNum field. **PBDTGetIconInfo** returns the size of the bitmap in ioDTActCount, the file type in ioFileType, and the icon size and color depth in ioIconType.

See the previous description of the **PBDTGetIcon** function for a list of values and their constants returned by **PBDTGetIconInfo** in the ioIconType field. Ignore any values that may be returned in ioIconType and that are not listed there; they represent special icons used only by the **Finder**.

To step through a list of the icon types supported by an application, make repeated calls to **PBDTGetIconInfo**, specifying a creator and an index value for ioIndex each call. Set the index to 1 on the first call, and increment it on each subsequent call until ioResult returns afpItemNotFound.

Returns: an Error code. It will be one of the following:

noErr	(0)	No error
ioErr	(-36)	I/O error
rfNumErr	(-51)	Reference number invalid
extFSErr	(-58)	External file system-file system identifier is nonzero
afpItemNotFound	(-5012)	Information not found

Note: There is a second, asynchronous, version of this function. It does not take a second parameter; instead, it adds the suffix "Async" to the name of the routine.

Similarly, the third (synchronous) version of the routine does not take a second parameter; instead, it adds the suffix "Sync" to the name of the routine.

Note, however, that the second and third versions of these routines do not use the glue code that the first versions use and are therefore more efficient.