PBDTAddIcon Add an icon to the desktop database.

#include <Files.h> **Finder Interface**

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

Your application should not ordinarily call the functions for adding data to the database. If your application does need to write to or delete information from the desktop database, it must call **PBDTFlush** to update the copy stored on the volume.

To add an icon definition to the desktop database, use the **PBDTAddIcon** function.

Parameter block

→	12	ioCompletion	long	completion routine
←	16	ioResult	short	result code
→	24	ioDTRefNum	short	desktop database reference number
→	28	ioTagInfo	long	reserved; must be initialized to 0
→	32	ioDTBuffer	long	pointer to icon data
→	36	ioDTReqCount	long	size of icon bitmap
→	45	ioIconType	char	icon type
→	52	ioFileCreator	long	icon's file creator
→	56	ioFileType	long	icon's file type

PBDTAddIcon adds an icon definition to the desktop database specified in ioDTRefNum. You specify the creator and file type that the icon is associated with in the ioFileCreator and ioFileType fields. For the icon type in ioIconType, specify either a constant or a value from the following list.

Constant	Value	Corresponding resource type	Description
kLargeIcon	1	'ICN#'	Large black-and-white icon with mask
kLarge4BitIcon	2	'icl4'	Large 4-bit color icon
kLarge8BitIcon	3	'icl8'	Large 8-bit color icon
kSmallIcon	4	'ics#'	Small black-and-white icon with mask
kSmall4BitIcon	5	'ics4'	Small 4-bit color icon
kSmall8BitIcon	6	'ics8'	Small 8-bit color icon

The value you supply in ioDTReqCount is the size in bytes of the buffer that you've allocated for the icon's bitmap pointed to by ioDTBuffer; this value depends on the icon type. Be sure to allocate enough storage for the icon data; 1024 bytes is the largest amount required for any icon under System 7.0. You can use a constant from the following list.

Constant	Value (bytes in bitmap)	Corresponding resource type	Description
kLargeIconSize	256	'ICN#'	Large black-and-white icon with mask
kLarge4BitIconSize	512	'icl4'	Large 4-bit color icon
kLarge8BitIconSize	1024	'icl8'	Large 8-bit color icon
kSmallIconSize	64	'ics#'	Small black-and-white icon with mask

kSmall4BitIconSize	128	'ics4'	Small 4-bit color icon
kSmall8BitIconSize	256	'ics8'	Small 8-bit color icon

You pass a pointer to the icon bitmap in the ioDTBuffer field. You must initialize the ioTagInfo field to 0.

If the database already contains an icon definition for an icon of that type, file type, and file creator, the new definition replaces the old.

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

noErr	(0)	No error
ioErr	(-36)	I/O error
wPrErr	(-44)	Volume is locked through hardware
vLckdErr	(-46)	Volume is locked through software
rfNumErr	(-51)	Reference number invalid
extFSErr	(-58)	External file system-file system identifier is nonzero
afpIconTypeError	(-5030	Sizes of new icon and one it replaces do not match

---

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.