PBCatSearch Search a volume's catalog

#include <Files.h> **File Manager**

OSErr	**PBCatSearch(***pb*, *async* **);**	
HParmBlkPtr	*pb* ;	address of a 76-byte CSParam structure
Boolean	*async* ;	0=await completion; 1=immediate return
	returns	Error Code; 0=no error

The **PBCatSearch** function searches a volume's catalog, using a set of search criteria that you specify. It builds a list of all files or directories that meet your specifications and returns it in an array of FSSpec records pointed to by the *ioMatchPtr* field of *pb*..

pb is the address of a CSParam structure.The relevant fields are as follows:

Out-In	Name	Type	Size	Offset	Description
→	ioCompletion	ProcPtr	4	12	pointer to completion routine
←	ioResult	short	2	16	result code
→	ioNamePtr	long	4	18	pointer to volume name
→	ioVRefNum	short	2	22	volume specification
→	ioMatchPtr	FSSpecPtr	4	24	pointer to array of to hold matches
→	ioReqMatchCount	long	4	28	maximum match count
←	ioActMatchCount	long	4	32	actual match count
→	ioSearchBits	long	4	36	enable bits for fields in criteria records
→	ioSearchInfo1	CInfoPBPtr	4	40	values and lower bounds
→	ioSearchInfo2	CInfoPBPtr	4	44	masks and upper bounds
→	ioSearchTime	long	4	48	maximum elapsed search time
→	ioCatPosition	16 bytes	16	52	current catalog position
→	ioOptBuffer	Ptr	4	68	pointer to optional read buffer
→	ioOptBufSize	long	I4	72	length of optional read buffer

async is a Boolean value. Use FALSE for normal (synchronous) operation or TRUE to enqueue the request and resume control immediately. See Async I/O.

Returns: an operating system Error Code. It will be one of:

noErr	(0)	No error (entire catalog has not been searched)
nsvErr	(-35)	Volume not found
ioErr	(-36)	I/O error
eofErr	(-39)	Logical end-of-file reached
paramErr	(-50)	Parameters don't specify an existing volume
extFSErr	(-58)	External file system
catalogChangedErr	(-1304)	Catalog has changed and catalog position record may be invalid

Notes: **PBCatSearch** searches the volume you specify for files or directories that match two coordinated sets of selection criteria

Field descriptions

ioCompletion A pointer to the completion routine or NIL.

ioResult Result code.

ioNamePtr A pointer to the name of the volume to be searched.

ioVRefNum The volume specification (volume reference number, working directory reference number, drive number, or 0 for default volume).

ioMatchPtr A pointer to an array where the file and directory names that match the selection criteria are returned. The array must be large enough to hold the largest possible number of **FSSpec** records, as determined by the ioReqMatchCount field.

ioReqMatchCount The maximum number of matches to return. This number should be the number of **FSSpec** records that will fit in the memory pointed to by ioMatchPtr. Use this field to avoid a possible excess of matches for criteria that prove to be too general.

ioActMatchCount The number of actual matches found.

ioSearchBits The fields of the parameter blocks ioSearchInfo1 and ioSearchInfo2 that are relevant to the search. See **Searching a Volume** under the section entitled **Using the File Manager** for the values of ioSearchBits.

ioSearchInfo1 A pointer to a **CInfoPBRec** parameter block that contains values and the lower bounds of ranges for the fields selected by ioSearchBits.

ioSearchInfo2 A pointer to a second **CInfoPBRec** parameter block that contains masks and upper bounds of ranges for the fields selected by ioSearchBits.

ioSearchTime A time limit on a search, in **Time Manager** format. Use this field to limit the run time of a single call to **PBCatSearch** . A value of 0 imposes no time limit.

ioCatPosition A position in the catalog where searching should begin. Use this field to keep an index into the catalog when breaking **PBCatSearch** down into a number of smaller searches. This field is valid whenever **PBCatSearch** exits because it either spends the maximum time allowed by ioSearchTime or finds the maximum number of matches allowed by ioReqMatchCount.

To start at the beginning of the catalog, set the initialize field of ioCatPosition to 0. Before exiting after an interrupted search, **PBCatSearch** sets that field to the next catalog entry to be searched. To resume where the previous call stopped, pass the entire **CatPositionRec** record returned by the previous call as input to the next.

ioOptBuffer A pointer to an optional read buffer. The ioOptBuffer and ioOptBufSize fields let you specify a part of memory as a read buffer, increasing search speed.

ioOptBufSize	The length of the buffer pointed to by ioOptBuffer. Buffer effectiveness varies with models and configurations, but a 16 KB buffer is likely to be optimal. Even a 1 KB buffer provides some performance improvement.

See **Searching a Volume** under the section **Using the File Manager** for a description of how to use **PBCatSearch** .

If the catalog changes between two timed calls to **PBCatSearch** (when you are using ioSearchTime and ioCatPosition to search a volume in segments and the catalog changes between searches), **PBCatSearch** returns a result code of catalogChangedErr. Depending on what has changed on the volume, ioCatPosition might be invalid, most likely by a few entries in one direction or another. You can continue the search, but you risk either skipping some entries or reading some twice.

When **PBCatSearch** has searched the entire catalog, it returns eofErr. If it exits because it either spends the maximum time allowed by ioSearchTime or finds the maximum number of matches allowed by ioReqMatchCount, it returns noErr.