DSPParamBlock structure

#include <ADSP.h>

typedef struct **DSPParamBlock** {

Size	Offset	Description
QElem *qLink;	4	0	Address of next queue entry; 0=end of queue
short qType;	2	4	Queue type
short ioTrap;	2	6	Routine trap
Ptr ioCmdAddr;	4	8	Routine address
ProcPtr ioCompletion;	4	12	Completion routine
OSErr ioResult;	2	16	Result code
StringPtr ioNamePtr;	4	18	Used only for dspOpen
short ioVRefNum;	2	22	Volume reference number
short ioCRefNum;	2	24	Driver reference number
short csCode	2	26	Primary command code
long qStatus;	4	28	Reserved for **ADSP**
short ccbRefNum;	2	32	CCBreference number
union {			
TRinitParams initParams;	26	34	dspInit, dspCLInit
TRopenParams openParams;	34	34	dspOpen, dspCLListen, dspCLDeny
TRcloseParams closeParams;	2	34	dspClose, dspRemove
TRioParams ioParams;	10	34	dspRead, dspWrite
TRattnParams attnParams;	10	34	dspAttention
TRstatusParams statusParams;	12	34	dspStatus
TRoptionParams optionParams;	6	34	dspOptions
TRnewcidParams newCID;	2	34	dspNewCID
} u;			
} **DSPParamBlock**;	134		

typedef DSPParamBlock ***DSPParamBlockPtr*** ;

Notes: The .DSP parameter block, defined by the **DSPParamBlock** data type, is a variant parameter block for the **PBControl** function.

The qLink, qType, ioTrap, ioCmdAddr, ioNamePtr, and ioVRefNum fields are filled in by the **Device Manager**; your application should not have to set or read these fields. The ioResult field returns the result of the function. If you call the routine asynchronously, the **Device Manager** sets this field to 1 as soon as you call the routine, and it changes the field to the actual result code when the routine completes execution. The ioCompletion field is a pointer to a completion routine that you can provide; the **Device Manager** calls your completion routine when it completes execution of the **PBControl** function. If you are not providing a completion routine, specify NIL for this field.

The ioCRefNum field is returned by the **OpenDriver** function. You must specify this number every time you call **The .DSP Driver**.

The csCode field specifies the command to be executed. You must fill in this field before calling the **PBControl** function. You can use the following constants as values for the csCode field:

ADSP routine selectors

```
dspInit       = 255;    {create a new connection end}
dspRemove     = 254;    {remove a connection end}
dspOpen       = 253;    {open a connection}
dspClose      = 252;    {close a connection}
dspCLInit     = 251;    {create a connection listener}
dspCLRemove   = 250;    {remove a connection listener}
dspCLListen   = 249;    {post a listener request}
dspCLDeny     = 248;    {deny an open-connection request}
dspStatus     = 247;    {get status of connection end}
dspRead       = 246;    {read data from the connection}
dspWrite      = 245;    {write data on the connection}
dspAttention  = 244;    {send an attention message}
dspOptions    = 243;    {set connection end options}
dspReset      = 242;    {forward reset the connection}
dspNewCID     = 241;{generate a cid for a connection end}
```

The qStatus field is reserved for use by **ADSP**. The ccbRefNum field is the reference number of the **CCB**. The **CCB** reference number is returned by **ADSP** in response to the dspInit routine. You must specify this number as a parameter to every .DSP driver routine you call subsequently.
The remaining fields are used only for specific routines; each of these fields is described in **.DSP Driver Routines**.