SEBlock structure

#include <Slots.h>

typedef struct **SEBlock** {		Size	Offset	Description
unsigned char	seSlot;	1	0	Slot number
unsigned char	sesRsrcId;	1	1	sResource ID
short	seStatus;	2	2	Status of code executed by sExec
unsigned char	seFlags;	1	4	Flags
unsigned char	seFiller0;	1	5	Filler, must be Signed Byte to align on odd boundary
unsigned char	seFiller1;	1	6	Filler
unsigned char	seFiller2;	1	7	Filler
long	seResult;	4	8	Result of sLoad
long	seIOFileName;	4	12	Pointer to IOFile name
unsigned char	seDevice;	1	16	Which device to read from
unsigned char	sePartition;	1	17	The partition
unsigned char	seOSType;	1	18	Type of OS
unsigned char	seReserved;	1	19	RefNum of the driver
unsigned char	seRefNum;	1	20	RefNum of the driver
unsigned char	seNumDevices;	1	21	Number of devices to load
unsigned char	seBootState ;	1	22	State of StartBoot code
} **SEBlock** ;		22		

typedef SEBlock ***SEBlockPtr**;

Notes: For the routine sExec, data transfer between the **Slot Manager** and card firmware takes place through this structure as well as through the **SpBlock** structure.