Sending a Message Block

Use the **PPCWrite** function to send a message block during a session specified by the session reference number.

You should call the **PPCWrite** function asynchronously. You can provide a completion routine that will be called when the **PPCWrite** function has completed, or you can poll the ioResult field of the PPC parameter block to detemine whether the **PPCWrite** function has completed. A **PPCWrite** completion routine can issue another **PPC Toolbox** call or set global variables. If another **PPC Toolbox** call is made from a completion routine, then the **PPCWrite** function must use a record of data type PPCParamBlockRec instead of type PPCWritePBRec. Note that message blocks are sent in the order in which they are written.

The following program illustrates how you use the **PPCWrite** function to write data during a session.

```
// Using the PPCWrite function to write data during a session

// Assuming inclusion of MacHeaders
#include <PPCToolBox.h>

// Prototype your write function like this prior to calling it
OSErr MyPPCWrite(PPCWritePBPtr,PPCSessRefNum,Size,Ptr);

OSErr MyPPCWrite( PPCWritePBPtr      thePPCWritePBPtr,
                  PPCSessRefNum      theSessRefNum,
                  Size               theBufferLength,
                  Ptr                theBufferPtr
                   )
{
    thePPCWritePBPtr->ioCompletion      = NULL;

    // from the PPCStart function or the PPCInform function--
    thePPCWritePBPtr->sessRefNum        = theSessRefNum;
    thePPCWritePBPtr->bufferLength      = theBufferLength;
    thePPCWritePBPtr->bufferPtr         = theBufferPtr;
    thePPCWritePBPtr->more              = FALSE;   // no more data to read
    thePPCWritePBPtr->userData          = 0;          // app-specific data
    thePPCWritePBPtr->blockCreator      = '????';  // app-specific data
    thePPCWritePBPtr->blockType         = '????';  // app-specific data

    return PPCWrite(thePPCWritePBPtr, TRUE);   // asynchronous
}
```

The first **PPCWrite** function that you use to create a new message block sets the block creator, block type, and user data attributes for the block. These attributes are returned to the application when it reads from the message block. Set the more field to FALSE to indicate the end of the message block or set this field to TRUE if you want to append additional data to a message block.

The following program illustrates a function that can be used to poll the ioResult field of a record of data type PPCWritePBRec. The function returns

TRUE when the PPCWrite function associated with PPCWritePBRec has completed.

```
// Polling the ioResult field to determine if a PPCWrite function has completed

// Assuming inclusion of MacHeaders
#include <PPCToolBox.h>

// Prototype your completion routine like this prior to calling it
Boolean MyWriteComplete(PPCWritePBPtr, OSErr *);

Boolean MyWriteComplete(PPCWritePBPtr thePPCWritePBPtr,OSErr *err)
{
    // Check ioResult for error
    *err = thePPCWritePBPtr->ioResult;

    // Return false if error is 1
    return !*err;
}
```