Handling Data During a Session

An application can both read from and write data to another application during a session. Use the **PPCRead** function during a session to read incoming blocks of data from another application.

Once a session is initiated, you should have a **PPCRead** function pending. You can issue a **PPCRead** function from inside a completion routine. This provides you with immediate notification if an error condition arises or the session closes.

The blockCreator, blockType, and userData fields are returned for the block you are reading. (These fields are set by the **PPCWrite** function.) To determine whether there is additional data to be read, check the more field. This field is FALSE to indicate the end of a message block.

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

```
// Using the PPCRead function to read data during a session

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

// Prototype your routine like this prior to calling it
OSErr MyPPCRead(PPCReadPBPtr,PPCSessRefNum,Size,Ptr);

OSErr MyPPCRead(PPCReadPBPtr        thePPCReadPBPtr,
                PPCSessRefNum       theSessRefNum,
                Size                theBufferLength,
                Ptr                 theBufferPtr
    )

{
    thePPCReadPBPtr->ioCompletion       = NULL;

    // from the PPCStart function  or the PPCInform function:
    thePPCReadPBPtr->sessRefNum         = theSessRefNum;
    thePPCReadPBPtr->bufferLength       = theBufferLength;
    thePPCReadPBPtr->bufferPtr          = theBufferPtr;

    return PPCRead(thePPCReadPBPtr, TRUE);    // asynchronous
}
```

You should make any calls to **PPCRead** asynchronously. You can provide a completion routine that will be called when the **PPCRead** function has completed, or you can poll the ioResult field of the PPC parameter block to determine whether the **PPCRead** function has completed. A **PPCRead** completion routine can issue another asynchronous **PPC Toolbox** call or set global variables. If another **PPC Toolbox** call is made from a completion routine, then the **PPCRead** function must use a record of data type PPCParamBlockRec instead of type PPCReadPBRec.

The following program illustrates a function that can be used to poll the

ioResult field of a record of data type PPCReadPBRec. The function returns TRUE when the **PPCRead** function associated with PPCReadPBRec has completed.

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

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

// Prototype your polling routine like this prior to calling it
Boolean MyReadComplete(PPCReadPBPtr,OSErr *);

Boolean MyReadComplete(PPCReadPBPtr thePPCReadPBPtr, OSErr *err)

{
   // Error result gets value of ioResult
    *err = thePPCReadPBPtr->ioResult;

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