Using the PPC Toolbox

To begin using the **PPC Toolbox**, you must determine whether it is available on the user's computer system by using the **Gestalt** function. The Gestalt selector is gestaltPPCToolboxAttr. A noErr result code indicates that the **PPC Toolbox** is present.

The **Gestalt** function returns a combination of the following constants: gestaltPPCToolboxPresent, gestaltPPCSupportsRealTime, gestaltPPCSupportsOutGoing, and gestaltPPCSupportsIncoming.

The **PPC Toolbox** currently supports only sessions in real time. The **Gestalt** function returns gestaltPPCSupportsRealTime by default. If this bit is not set, you need to initialize the **PPC Toolbox**.

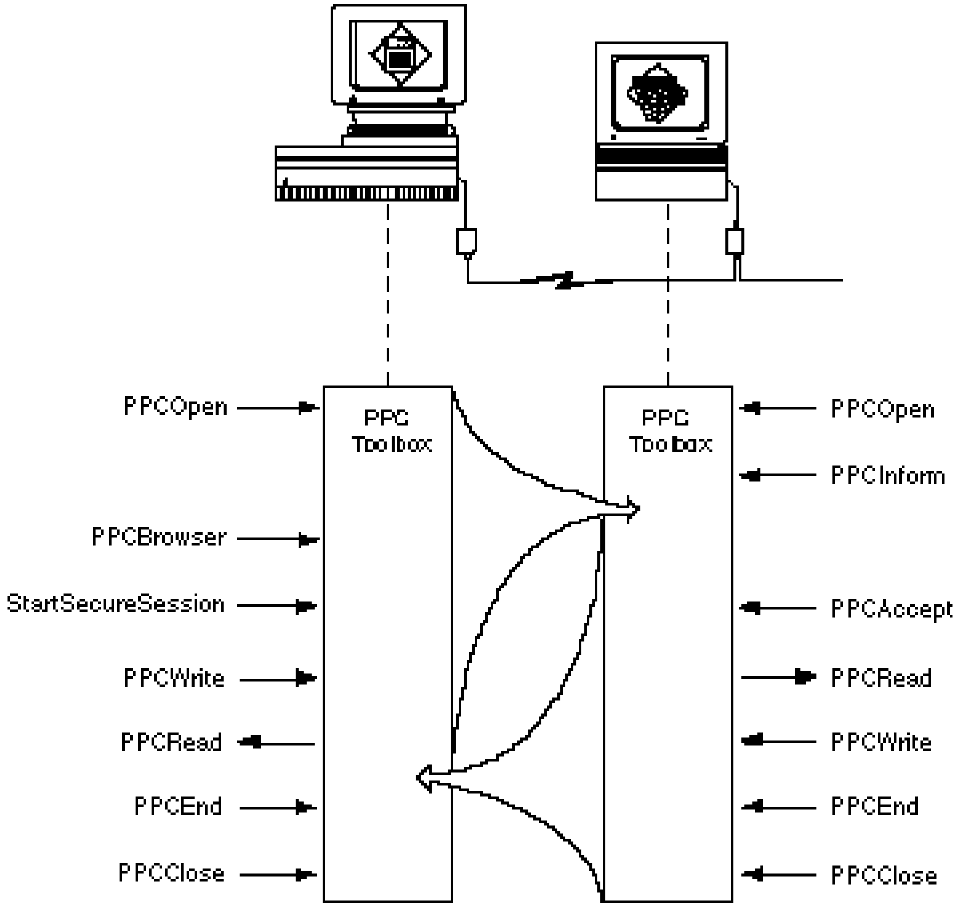
The **Gestalt** function returns gestaltPPCSupportsOutGoing to indicate support of outgoing sessions across a network of Macintosh computers. If this bit is not set, the user hasn't enabled **AppleTalk** in the Chooser.

The **Gestalt** function returns gestaltPPCSupportsIncoming if the user has enabled program linking in the Sharing Setup control panel. If this bit is not set, the user either hasn't enabled **AppleTalk** in the Chooser or hasn't enabled program linking in the Sharing Setup control panel.

The following program illustrates how you use the **PPCInit** function to initialize the **PPC Toolbox**:

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

// Prototype your initialization routine like this prior to calling it
OSErr MyPPCInit(void);
OSErr  MyPPCInit()

{
    long PPCAttributes;   // Attributes to set
    OSErr err;            // Error returned

    err = Gestalt(gestaltPPCToolboxAttr, &PPCAttributes);
    if ( err == noErr )
    {
        // PPC Toolbox is present
        if (  !( PPCAttributes & gestaltPPCSupportsRealTime) )
        {
            // PPC Toolbox needs initialization
            // initialize the PPC Toolbox and set function result
            err = PPCInit();
            // test the attributes for the PPC Toolbox
            err = Gestalt(gestaltPPCToolboxAttr,&PPCAttributes);
        }

        if ( PPCAttributes & gestaltPPCSupportsOutGoing )
            ;
            // ports can be opened to the outside world
        else
            ;
            // it's likely that AppleTalk is disabled, so you may
            // want to tell the user to activate AppleTalk from
            // the Chooser

        if (PPCAttributes & gestaltPPCSupportsIncoming)
            ;
            // ports can be opened with location names that the
            // outside world can see
        else
            ;
            // it's likely that program linking is disabled, so you
            // may want to tell the user to start program linking
            // from the Sharing Setup control panel
    }
    return err;
}
```

The following figure illustrates a database application (on the left) that has initiated a session with a spreadsheet application (on the right) to exchange data using the **PPC Toolbox**. This figure includes an example of the sequence of **PPC Toolbox** routines executed by these applications. Detailed descriptions of the functions appear in the sections that follow.



Database and spreadsheet applications using the **PPC Toolbox**

To establish a session, each application must first open a port using the

PPCOpen function. The spreadsheet application prepares to receive session requests by calling the **PPCInform** function.

Before initiating a session or opening a port, the database application can let the user browse through the list of available ports (using the **PPCBrowser** function). If the user decides to communicate with the spreadsheet application, the database application initiates a session with the spreadsheet application's port using the **StartSecureSession** function. After the **PPC Toolbox** authenticates the user name and password of the initiating port, the spreadsheet application accepts the session request (using the **PPCAccept** function).

Once the session is established, the applications exchange information in the form of message blocks (using the **PPCRead** and **PPCWrite** functions). During a session, an application can both read from and write message blocks to another application. After the information exchange is done, each application ends the session (**PPCEnd**) and then closes its port (**PPCClose**) when it quits.

The **PPCOpen** function returns a port reference number. The port reference number is a reference number for the port through which you are requesting a session. The database application uses the port reference number in subsequent calls to the **StartSecureSession** and **PPCClose** functions. The **StartSecureSession** function returns a **session reference number**. The session reference number is used to identify the session during the exchange of data. It is used in subsequent calls to the **PPCWrite**, **PPCRead,** and **PPCEnd** functions.

The **PPCOpen** function returns a port reference number that the spreadsheet uses in subsequent calls to the **PPCInform** and **PPCClose** functions. The **PPCInform** function returns a session reference number that is used in subsequent calls to the **PPCAccept**, **PPCRead**, **PPCWrite**, and **PPCEnd** functions.