Preparing for a Session

To communicate, you can open a port for your application and make it available to receive session requests, to initiate sessions, or both. Applications that are able to receive session requests can choose to accept or reject incoming session requests.

Before an application can accept and establish a session with another application, the **PPC Toolbox** authenticates the initiating user (unless guest access is enabled or the applications are located on the same computer). Once a session begins, the two applications can exchange data with each other.

Initiating a PPC Session

Once you have established the name and the location of the port that you want to communicate with, you can initiate a session. You can use either the **StartSecureSession** function or the **PPCStart** function to initiate a session. The **StartSecureSession** function displays several dialog boxes on the user's screen to identify each user who requests a session. You may prefer to use the **PPCStart** function for low-level code such as that used for drivers, which typically do not provide a user interface. You may also prefer to use **PPCStart** when the application you are initiating a session with does not require authentication. The **IPCListPorts** and **PPCBrowser** functions return information about whether a particular port requires authentication.

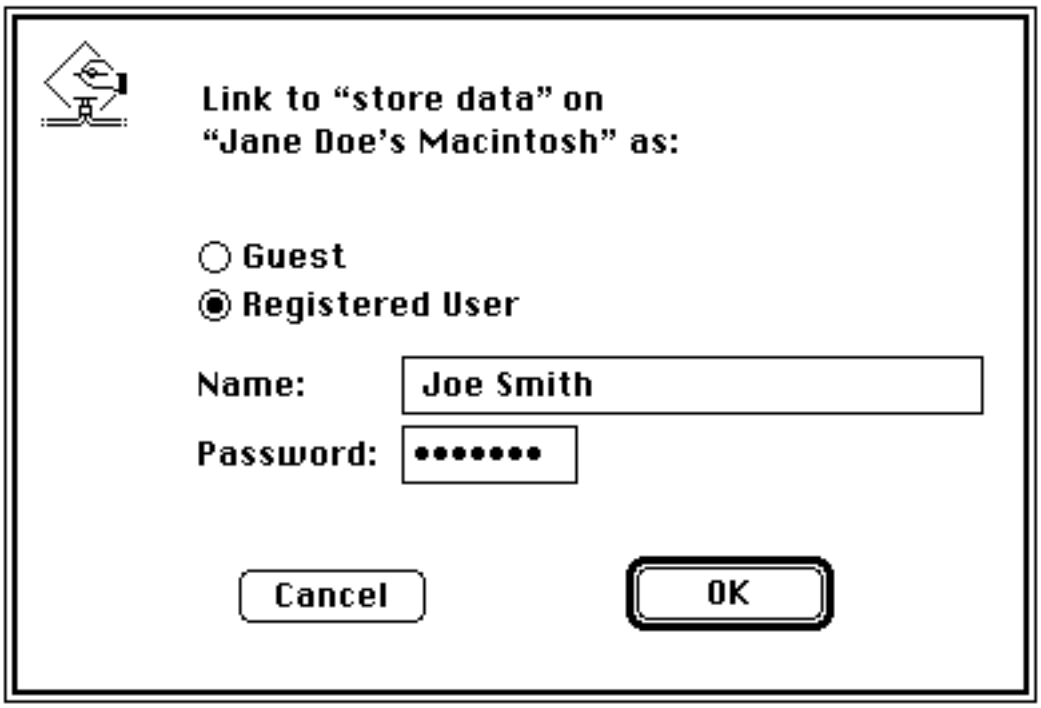
Note: Do not call the **StartSecureSession** function from an application that is running in the background, since it requires that several dialog boxes appear on the user's screen.

The **StartSecureSession** function provides authentication services to identify each user who requests a session. This function combines the processes of prompting for user name and password and initiating a session into one synchronous procedure call. If authentication fails, the **PPC Toolbox** rejects the incoming session request.

```
err = StartSecureSession (&pb, &userName, useDefault, allowGuest,
                          guestSelected, prompt);
```

Set the useDefault parameter to TRUE if you want the **StartSecureSession** function to use the default user identity (described later in this section). If the default user identity cannot be authenticated, the **StartSecureSession** function displays a dialog box to allow a user to log on. The following picture shows the user identity dialog box.

THINK Reference © 1991-1992 Symantec Corporation



The user identity dialog box

The prompt parameter of the **StartSecureSession** function allows you to specify a line of text that the dialog box can display. The allowGuest parameter specifies whether to enable the Guest radio button. If a port requires authentication, you should set this parameter to FALSE.

The userName parameter specifies the name of the user who is attempting to initiate a session. If the user name is not specified, the user identity dialog box appears on the user's screen with the owner name provided from the Sharing Setup control panel.

If the user enters an invalid password, the **StartSecureSession** function displays the dialog box shown in the next picture:

The incorrect password dialog box

After the user clicks OK, the user identity dialog box reappears in the foreground so that the user can enter the password again.

THINK Reference © 1991-1992 Symantec Corporation

If the user's name is invalid, the **StartSecureSession** function displays the dialog box shown in the following picture:

The invalid user name dialog box

After the user clicks OK, the user identity dialog box reappears so that the user can enter a new user name.

The **StartSecureSession** function remains in this loop until a secure session is initiated or the user clicks Cancel in the user identity dialog box. If a secure session is initiated, **StartSecureSession** returns the user reference number in the corresponding field in the **PPCStart** parameter block. The user reference number represents the user name and password. A user reference number of 0 indicates that a session has been initiated with guest access.

Before your application quits, you need to invalidate all user reference numbers obtained with the **StartSecureSession** function except for the default user reference number and the guest reference number (0).

The following program illustrates how to use the **StartSecureSession** function to establish an authenticated session. This listing shows only one session, although your application may conduct multiple sessions at one time.

```
// Using the StartSecureSession function to establish a session
// Assuming inclusion of macheaders
#include <PPCToolBox.h>

// Prototype code like this prior to calling it:
OSErr MyStartSecureSession
            (PortInfoPtr,
            LocationNamePtr,
            PPCPortRefNum,
            PPCSessRefNum *,
            long *,  long *,  Str32,  Boolean *);

OSErr MyStartSecureSession
            (PortInfoPtr            thePortInfoPtr,
            LocationNamePtr         theLocationNamePtr,
            PPCPortRefNum           thePortRefNum,
```

```
                PPCSessRefNum       *theSessRefNum,
                long                *theUserRefNum,
                long                *theRejectInfo,
                Str32               userName,
                Boolean             *guestSelected)
{
    PPCStartPBRec thePPCStartPBRec;
    Boolean     useDefault;
    Boolean     allowGuest;
    OSErr       err;

    thePPCStartPBRec.ioCompletion = NULL;

    thePPCStartPBRec.portRefNum = thePortRefNum;
                                        // from the PPCOpen function
    thePPCStartPBRec.serviceType = ppcServiceRealTime;

    thePPCStartPBRec.resFlag = 0;

    thePPCStartPBRec.portName = &thePortInfoPtr->name;
                                        // from PPCBrowser function

    thePPCStartPBRec.locationName = theLocationNamePtr;
                                        // from PPCBrowser function
    thePPCStartPBRec.userData = 0;


    // try to connect with default user identity
    useDefault = TRUE;

    // highlight the Guest button appropriately
    allowGuest = !thePortInfoPtr->authRequired;

    err = StartSecureSession(&thePPCStartPBRec, userName, useDefault,
                                allowGuest, guestSelected, NULL);
    if ( err == noErr ) {
        *theSessRefNum = thePPCStartPBRec.sessRefNum;
        *theUserRefNum = thePPCStartPBRec.userRefNum;
    }
    else if (err == userRejectErr)
        // return rejectInfo from the PPCReject function
        *theRejectInfo = thePPCStartPBRec.rejectInfo;
    return err;
}
```

For low-level code such as that used for drivers (which typically do not provide a user interface), you can use the **PPCStart** function instead of the **StartSecureSession** function to initiate a session. You can also use the **IPCListPorts** function (instead of displaying the program linking dialog box) to obtain a list of ports.

If the authRequired field of the port information record contains FALSE, the port allows guest access. If the authRequired field of the port information record contains TRUE, use the **PPCStart** function and the user reference number obtained previously from the **StartSecureSession** function to

THINK Reference © 1991-1992 Symantec Corporation

reestablish an authenticated session.

You can also attempt to log on as the default user using the **GetDefaultUser** function to obtain the default user reference number and the default userName. The default userName is established after the owner starts up the computer.

```
err = GetDefaultUser (&userRef,  userName);
```

The userRef parameter is a reference number that represents the user name and password of the default user. The userName parameter contains the owner name that is specified in the Sharing Setup control panel.

The **GetDefaultUser** function returns an error when the default user identity does not exist (no name is specified in the Sharing Setup control panel) or the user is not currently logged on.

The following program illustrates how you use the **PPCStart** function to initiate a session. The **PPCStart** function uses the port information record and the location name record to attempt to open a session with the selected PPC port.

```
// Initiating a session using the PPCStart function
// Assuming inclusion of MacHeaders
#include <PPCToolBox.h>

// Prototype your function like this prior to calling it
OSErr MyPPCStart(PortInfoPtr,  LocationNamePtr,  PPCPortRefNum,
                 PPCSessRefNum *,  long *,  long *);

OSErr MyPPCStart    (PortInfoPtr        thePortInfoPtr,
                     LocationNamePtr    theLocationNamePtr,
                     PPCPortRefNum      thePortRefNum,
                     PPCSessRefNum      *theSessRefNum,
                     long               *theUserRefNum,
                     long               *theRejectInfo)
{
    PPCStartPBRec   thePPCStartPBRec;
    Str32           userName;
    OSErr           err;

    thePPCStartPBRec.ioCompletion       = NULL;

    thePPCStartPBRec.portRefNum         = thePortRefNum;
                                        // from PPCOpen function

    thePPCStartPBRec.serviceType        = ppcServiceRealTime;

    thePPCStartPBRec.resFlag            = 0;

    thePPCStartPBRec.portName           = &thePortInfoPtr->name;
                                        // destination port

    thePPCStartPBRec.locationName       = theLocationNamePtr;
                                        // destination location
```

THINK Reference © 1991-1992 Symantec Corporation

```
// application-specific data for PPCInform
thePPCStartPBRec.userData = 0;

    err = GetDefaultUser(&thePPCStartPBRec.userRefNum, userName);
    if ( err != noErr )
      thePPCStartPBRec.userRefNum = 0;

    if ( thePortInfoPtr->authRequired && !thePPCStartPBRec.userRefNum )
      // port selected does not allow guests and you do not have a default
      // user reference number, so you cannot log on to this port
       err = authFailErr;
    else
      // attempt to log on
       err = PPCStart(&thePPCStartPBRec, FALSE);

    if ( err == noErr ) {
      *theSessRefNum  = thePPCStartPBRec.sessRefNum;
      *theUserRefNum = thePPCStartPBRec.userRefNum;
    }
    else if ( err == userRejectErr )
        // return rejectInfo from the PPCReject function
        *theRejectInfo = thePPCStartPBRec.rejectInfo;
    return err;
}
```

The port to which you wish to connect must have an outstanding **PPCInform** function to successfully start a session. You cannot initiate a session with a port that is not able to receive session requests.

If the port is open, has an outstanding **PPCInform** posted, and accepts your session request, the **PPCStart** function returns a noErr result code and a valid session reference number. This session reference number is used to identify the session during the exchange of data.

THINK Reference © 1991-1992 Symantec Corporation