Generating a List of Ports

To generate a list of ports without displaying dialog boxes, you can use the **IPCListPorts** function. The **IPCListPorts** function allows you to obtain a list of ports on a particular computer within a particular zone. To obtain a list of ports, several steps are required. First, use the **GetZoneList** function to obtain a list of zones. Next, you must use the **PLookupName** function to obtain a list of computers with ports. After establishing the zone and the computer, you can use the **IPCListPorts** function to obtain the list of available ports.

The following program illustrates how you use the **IPCListPorts** function to obtain a list of ports on a particular computer. This function returns a list of port information records in the buffer pointed to by the parameter thePortInfoBufferPtr. The actual number of port information records is returned in the parameter theActualCount.

```
// Using the IPCListPorts function to obtain a list of ports
// Assuming Inclusion of MacHeaders
#include <PPCToolBox.h>
#include <Script.h>

// include string.h for strcpy function
#include <string.h>

// Prototype your function like this prior to calling it
OSErr MyIPCListPorts(short,short,short *,Str32,Str32,PortInfoArrayPtr);

OSErr MyIPCListPorts(short theStartIndex,
    short   theRequestCount, short    *theActualCount,
    Str32  theObjStr, Str32  theZoneStr,
    PortInfoArrayPtr thePortInfoBufferPtr )

{
    IPCListPortsPBRec     theIPCListPortsPBRec;
    PPCPortRec    thePPCPortRec;
    LocationNameRec   theLocationNameRec;
    OSErr  err;

    // list all PPC ports at the specified location
    thePPCPortRec.nameScript = smRoman;

    // set name to match all names, NOTE call to CtoPstr to convert string
    // back to Pascal style string.
    strcpy((char *)thePPCPortRec.name,"=");
    CtoPstr(thePPCPortRec.name);

    thePPCPortRec.portKindSelector = ppcByString;

    // match all types,  NOTE call to CtoPstr to convert string
    // back to pascal style string after casting for call to strcpy
    strcpy((char *)thePPCPortRec.u.portTypeStr,"=");
    CtoPstr(thePPCPortRec.u.portTypeStr);

    // The application must choose and supply the NBP zone string from
    // the list returned by GetZoneList. Then, the application must
    // choose and supply the NBP object string from the list returned by
    // NBPLookup. This example looks for NBP type "PPC Example". If you
    // don't supply your own NBP type, you should use "PPCToolBox" for
    // the NBP type string.

    theLocationNameRec.locationKindSelector = ppcNBPLocation;

    // copy objstr, note casting to make C type string for strcpy
    // then convert string back to pascal type string
    strcpy ((char *)theLocationNameRec.u.nbpEntity.objStr,
            (char *)theObjStr);
    CtoPstr(theLocationNameRec.u.nbpEntity.objStr);

    // copy typestr, note casting to make C type string for strcpy
    // then convert string back to pascal type string
    strcpy((char *)theLocationNameRec.u.nbpEntity.typeStr,"PPC Example");
    CtoPstr(theLocationNameRec.u.nbpEntity.typeStr);

    // copy zonestr, note casting to make C type string for strcpy
    // then convert string back to pascal type string
    strcpy((char *)theLocationNameRec.u.nbpEntity.zoneStr,
            (char *)theZoneStr);
    CtoPstr(theLocationNameRec.u.nbpEntity.zoneStr);

    theIPCListPortsPBRec.startIndex     = theStartIndex;
    theIPCListPortsPBRec.requestCount   = theRequestCount;
    theIPCListPortsPBRec.portName       = &thePPCPortRec;
    theIPCListPortsPBRec.locationName   = &theLocationNameRec;
    theIPCListPortsPBRec.bufferPtr      = thePortInfoBufferPtr;

    err = IPCListPorts(&theIPCListPortsPBRec, FALSE);
    *theActualCount = theIPCListPortsPBRec.actualCount;
    return err;
}
```

The **IPCListPorts** function returns information about ports that are on the computer specified in the locationName field of the list ports parameter block. If the locationName field is NIL or if the locationKindSelector field in the location name record is ppcNoLocation, the **IPCListPorts** function returns only the port names for the local computer.

The bufferPtr field points to an area of memory that contains the requested port names. You are responsible for allocating enough memory to hold the requested port names. The buffer length must be equal to

```
sizeof(PortInfoRec) * requestCount
```