

How to Get Zone Information Operating instructions

The Zone Information Protocol (ZIP) obtains the zone information by using the AppleTalk Transaction Protocol (ATP) to send an information request to a router. The xppTimeout field specifies the amount of time, in seconds, that **The .ATP Driver** should wait between attempts to obtain the data. The xppRetry field specifies the number of times **The .ATP Driver** should attempt to obtain the data before returning the reqFailed (request failed) result code.

The zipBuffPtr field is a pointer to a data buffer that you must allocate. This buffer must be 578 bytes for the **GetZoneList** and **GetLocalZones** functions and 33 bytes for the **GetMyZone** function. ZIP returns the zone names (as a packed array of packed strings) into this buffer. The zipNumZones field returns the actual number of zone names that ZIP placed in the buffer. You must set the zipLastFlag field to 0 (FALSE) before you execute the **GetZoneList** or **GetLocalZones** function. If the zipLastFlag parameter is still 0 when the command has completed execution, then ZIP is waiting to return more zone names. In this case you must empty the buffer (or allocate a new one) and call the **GetZoneList** or **GetLocalZones** function again immediately. When there are no more zone names to return, ZIP sets the zipLastFlag field to a nonzero (TRUE) value.

The zipInfoField field is a 70-byte data buffer that you must allocate for use by **The .XPP Driver**. The first time you call any of these functions, you must set the first word of this field to 0. You must not change any values in this field subsequently.

The listing below illustrates the use of the **GetZoneList** function. The **GetLocalZones** function operates in exactly the same fashion.

```
// Using the GetZoneList function
// Assuming inclusion of <MacHeaders>

#include <AppleTalk.h>

void doGetZoneList (void);
void EmptyDataBuf(XPPParamBlock *myXPPPB);
void YourZIPProc(XPPParamBlock *myXPPPB);
void DoError (OSErr myErr);

void doGetZoneList ()
{
    XPPParamBlock  myXPPPB;// .XPP parameter block
    OSErr  myErr;

    myXPPPB.XCALL.xppTimeout = 3;   // timeout period for .XPP
    myXPPPB.XCALL.xppRetry = 4;   // retry count
    myXPPPB.XCALL.zipBuffPtr = NewPtr(578);// zone names returned here
    myXPPPB.XCALL.zipLastFlag = 0;   // set to 0 the first time through
    myXPPPB.XCALL.zipInfoField[0] = 0;
    myXPPPB.XCALL.zipInfoField[1] = 0;   // first word is 0 the first
                                         // time through
}
```

```
myErr = noErr;

// Check the zipNumZones field to determine how many zone names
// have been returned in the buffer. Append the zone names to the
// end of your own buffer before returning to read more zone names.

// loop to get all of the zone names

while ( (!myXPPPB.XCALL.zipLastFlag) && (!myErr) ) {

    myErr = GetZoneList(&myXPPPB, FALSE);
    EmptyDataBuf(&myXPPPB); // your routine to empty
    // data buffer
}
YourZIPProc(&myXPPPB); // your routine to process names
if (myErr != noErr)
    DoError(myErr);

// there's an error
DisposPtr(myXPPPB.XCALL.zipBuffPtr); // give space back
}
```

Assembly-language note: The .XPP Driver functions all use the same value (xCall, which is equal to 246) for the csCode parameter to the XPP parameter block. The xCall routine uses the value of the xppSubCode parameter to distinguish between the functions, as follows:.

Function	<u>xppSubCode</u>	Value
<u>GetMyZone</u>	zipGetMyZone	7
<u>GetLocalZones</u>	zipGetLocalZones	5
<u>GetZoneList</u>	zipGetZoneList	6