



Tech Info Library

MPW 3.0: Problem Using C Compiler

Revised: 12/4/89
Security: Everyone

MPW 3.0: Problem Using C Compiler

=====

This article last reviewed: 25 October 1989

TOPIC -----

I think that there is a problem in "AppleTalk.h". It seems that the C compiler doesn't feel that the definition of the toolbox routine "BuildDDPwds()" is compatible with the definition of "AddrBlock". It seems like the actual toolbox routine is expecting the parameter to be passed by (a scalar) value, but the compiler is expecting a pointer.

I could not make one of our programs compile and work correctly until I changed the "AddrBlock" parameter to a type "long" in "AppleTalk.h" and passed the AddrBlock parameter in as a type "long" scalar parameter. Here is my change to "AppleTalk.h":

```
#if 0
pascal void BuildDDPwds(Ptr wdsPtr,Ptr headerPtr,Ptr dataPtr,const AddrBlock
*netAddr,short ddpType,short dataLen);
#else
pascal void BuildDDPwds(Ptr wdsPtr,Ptr headerPtr,Ptr dataPtr,long myAddrBlock,
short ddpType,short dataLen);
#endif
```

Any comments?

DISCUSSION -----

Although it seems at first that there is an error in the way this particular parameter is declared, the declaration actually makes sense when you look at the underlying mechanisms used by Macintosh Toolbox calls.

Because Macintosh Toolbox calls use Pascal parameter-passing conventions, an AddrBlock parameter is always passed as the address to that parameter. This happens because any parameter longer than 4 bytes automatically has its address passed to preserve stack space.

To mimic this calling convention, the C declaration for the BuildDDPwds declares the AddrBlock parameter as a pointer to an AddrBlock structure. When you call the BuildDDPwds routine, you are expected to pass a pointer to an AddrBlock structure you have declared.

For more information, see the Parameter types section of the "MPW C Reference."

Copyright 1989 Apple Computer, Inc.

Tech Info Library Article Number:4717