<u>MacTCP 1.1 Header File Incompatibilities and Fixes</u>

MacTCP 1.1, Apple's System7 compatible version of MacTCP fixes several problems in earlier releases, but failed to address several incompatibilities in the header files.

This short note outlines the problems with the 1.1 header files, but does not include the fixes. Patches in the form of MPW "compare" output are available on Apple's Developer CD series. Apple cannot provide the new headers directly, as they are only available as part of the "MacTCP Developers' Kit", available from APDA (part #M0704/C).

The problems with the headers can be grouped into three areas: General bugs or shortcomings, Think C incompatibilities, and C++ incompatibilities. I have listed the header problems below, organized by the file containing the problem and classification of the problem.

NOTE: These changes have not been thoroughly tested, so you will have to use them at your own risk.

<u>All Files</u>

(1) General Problem
Each header file is missing the:

```
#ifdef __cplusplus
extern "C" {
#endif
```

and:

```
#ifdef __cplusplus
extern "C" {
#endif
```

which are necessary for C++ unmangling.

(2) General Problem
There are no #ifdefs to prevent including a file multiple times, as in other Apple headers. This was fixed by adding the following to each of the header files:

```
#ifndef __MACTCPCOMMONTYPES__
#define __MACTCPCOMMONTYPES__
...
#endif
```

where "MACTCPCOMMONTYPES" is replaced by the name of the header file.
(2a) Think C Problem

Think C before 5.0 does not support pascal typedefs, which MacTCP makes use of, so #ifdef THINK_C was added in several instances to declare these pascal typedef functions as type ProcPtr. This fixes the problem.

<u>GetMyIPAddr.h</u>

(3) General Problem
ParamBlockHeader is used as a #define here, conflicting with its use in <Files.h>. The solution is to change the name of the #define to IPParamBlockHeader.

<u>MiscIPPB.h</u>

(4) General Problem
Same as (3) above. In this case, it was changed to GetIPParamBlockHeader.

(5) General Problem
The structure IPParamBlock is defined in this file with a different definition from the one in GetMyIPAddr.h. The solution is to change the name of the struct.

(6) General Problem
AppleTalk.h is required for successful compilation. Code was added to include this if it has not already been included.

(7) General Problem
icmpEchoTimeoutErr is defined in MacTCPCommonTypes in addition to being defined in MiscIPPB.h. Solution is to remove its definition here.

<u>AddressXLation.h</u>

(8) Think C Incompatibility
The “int” type is used within the “hostInfo” structure definition. The default size of an int is different from Think to MPW. In all cases, int in the MacTCP headers should be changed to long.

(9)Think C Incompatibility
The returnRec struct uses an int. Fix is to change to a long as above.

(10) Think C Incompatibility
The CloseResolver() prototype is defined with an empty parameter list, not with a void parameter list. Think C requires the void to consider the definition a prototype. The fix is to add the void keyword as the function parameter list.
(11) C++ Incompatibility
The cacheEntryRecord struct uses a variable named “class”. This causes major problems with C++ compilers, since the class keyword takes on a different meaning in this instance.

The fix is to change this to “cacheClass”.

<u>dnr.c</u>

(12) General/Think C Problem
The typedef for OSErrProcPtr was incomplete, causing ambiguity for argument casting. This was chagned to "`typedef OSErr (*OSErrProcPtr)(long,...);`" to make sure the first parameter to the name resolver calls is passed as a long.

(13) Think C Incompatibility
Defines in Think default to short, not long, so all name resolver calls are made incorrectly. This is fixed by the typedef in (12), but Think 4.0.5 does not recognize partial prototypes correctly. The fix is to add a “L” to the end of each of the name resolver command #defines.

(14) Think C Incompatibility
The MXInfo() procedure has an extra semicolon after the trailing brace “};” which causes a syntax error with the Think C compiler. The semicolon was removed to fix the problem.