

# New Technical Notes

Macintosh



®

---

Developer Support

## Strip With `_OpenResFile` and `_OpenRFPPerm` Toolbox

Written by: Byron Han

April 1989

This Technical Note discusses a bug in `_OpenResFile` and `_OpenRFPPerm` which can cause system crashes and what you can do to avoid this problem.

---

The traps `_OpenResFile` and `_OpenRFPPerm` call some common code in 128K and later ROMs which was affected by some system patches for early print drivers. The problem is that the common code checks an attribute bit in the pointer to the string name to see if it is a dereferenced handle. If the pointer has the resource attribute bit set, the Resource Manager assumes that it is a dereferenced handle and calls `_RecoverHandle`. This usually works, but when the string is embedded in a code resource, the Resource Manager calls `_RecoverHandle` with an invalid master pointer.

**Note:** In MPW C, this bug is **not** a problem, unless you use either the `-b`, `-b2`, or `-b3` options, which embed string constants in the code segment. If you use these options, you must deal with this bug.

The following code fragments give an example of this bug:

### MPW Pascal

```
VAR
    fileName : Str255;
    ref : INTEGER;

BEGIN
    fileName := 'This File';
    ref := OpenResFile(fileName);
END
```

### MPW C

```
Str255    fileName;
short int ref;

fileName = 'This File';
ref = OpenResFile(fileName);
```

Calling `_StripAddress` on the pointer to the filename prior to calling `_OpenResFile` or `_OpenRFPPerm` solves the problem:

### MPW Pascal

```
VAR
    fileName : Str255;
    ref : INTEGER;

BEGIN
    fileName := 'This File';
    ref := OpenResFile(StringPtr(StripAddress(@fileName))^);
END;
```

### MPW C

```
Str255      fileName;
short int   ref;

fileName = 'This File';
ref = OpenResFile((StringPtr)StripAddress((Ptr)fileName));
```

By always calling `_StripAddress` before calling `_OpenResFile` or `_OpenRFParm`, you will not have to deal with this problem, life will be good, and you will be able to rest a bit easier.

### Further Reference:

---

- M.OV.32BitClean
- M.ME.StripAddress