
Subject: RE: OBJ/COM formats
Posted by [Anonymous](#) on Mon, 27 May 2013 01:03:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Originally posted by: markz@microsoft.UUCP (Mark Zbikowski)

Message-ID:
Date: Mon, 13-Aug-84 18:28:33 EDT
Article-I.D.: microsoft.8707
Posted: Mon Aug 13 18:28:33 1984
Date-Received: Wed, 15-Aug-84 01:39:38 EDT
Organization: Microsoft Corporation
Lines: 37

"... I am not including this statement because with it exe2bin would not convert the file..."

EXE2BIN will convert an EXE file to a COM-format file under the following circumstances:

- o Entry point is 0:100h
- o No Stack segment is present
- o Image size is < 65536-100h-2
- o There are NO relocatable long references.

I strongly suspect that there is something else at work here.

"... I suspect that it is something to do with having data segments separate from the code segment..."

You have probably just hit the nail on the head. If you are using the macro assembler from IBM, you'll find that it will emit segment definitions in alphabetic order rather than in declaration order. This *may* be causing your data segment to precede your code segment which might violate the first condition above.

I really question your need of COM files in the first place. The *only* place that they might be useful is in device drivers. Outside of these, MSDOS is quite capable of performing all relocation for you. I strongly urge anyone who is thinking of using COM files to stick with the EXE format. Reasons:

- o No need for EXE2BIN.
- o Full use of instruction set (you can't effectively use the long call instruction, for example) by being relocatable.
- o Has extensible header should any changes to the format be needed.

- o EXE files don't have 64K-100h-2 size restriction.
 - o Full control of stack placement. COM files have no guarantee on minimal size of stack. I have seen a great numbers of COM files that fail on small-memory machines or on reading large data over where their stack was.
-