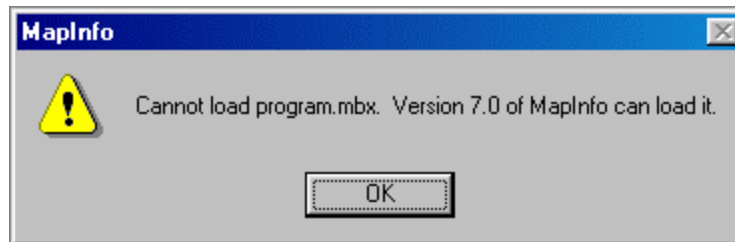


# MBXVER: MapBasic Executable Version Fixer

© Copyright 1996-2002 Stopwatch Maps

Have you ever tried to run a MapBasic program, only to have MapInfo reject your request with this message?



Perhaps you are running MapInfo version 6.5 but the MapBasic program was compiled under version 7.0 of MapBasic. MapInfo refuses to load and run a MapBasic executable compiled under a version *later* than its own version, for fear that the program *might* call upon facilities that *this* version of MapInfo can not perform ... even if the program doesn't do that at all.

This is frustrating for any user, and it's frustrating for the program developer who wants to compile using the latest version of MapBasic, even if the program being compiled calls only on facilities that were available in an earlier version of MapInfo. There needs to be a way to allow a program compiled under a later version of MapBasic to run under an earlier version of MapInfo *if* that program doesn't call upon facilities beyond the earlier version of MapInfo. As we will see, there is a way.

From what we've said, you would correctly guess that the MapBasic compiler always embeds, in any MapBasic executable (.MBX) file it generates, its *own* version number, *regardless* of the level of MapInfo facilities the program calls on. That mismatch is what leads to the error situation we started with. But now imagine that, after a MapBasic program has been compiled, we could:

- Check the facilities the MapBasic program calls on, to determine the *earliest* version of MapInfo that can execute this MapBasic program; then ...
- Change the version number in the .MBX file to indicate *that* version to be the MapInfo version required

MapInfo won't run a MapBasic program with a version number later than its own. But it *will* run a MapBasic program with a version number *earlier* than its own. So by performing the special actions we just described, our MapBasic program -- with its own version modified -- would run with any version of MapInfo that could logically support it.

We offer two versions of a program that performs this corrective action:

- A command-line version, MBXVER ... Easy to use, and it can be run from a DOS batch file ... but it provides only its help text as guidance
- A windowed version, MBXVERSION ... Interactive, and can not be batched ... But easier for the first-time user

## *The Command-Line Program*

The program **MBXVER.EXE** is a DOS program that examines the MapInfo facilities called on in a MapBasic executable program, determines the earliest MapInfo version that supports all those facilities, reports this and – if you tell it to – *sets* that version number into the MBX program. The DOS command has this syntax:

**MBXVER** *MBX\_filespec* [/V]

The file specification of the MBX file is required (the **.MBX** extension does not have to be specified). The program reports the version *currently* specified in the MBX file and the version that it believes is the *earliest* version of MapInfo that can run that program. If you specify the optional /V, it also *sets* that version number into the MBX file, allowing any version of MapInfo from that version forward to run this MapBasic program.

If you wish to *force* the version number in the MBX file, regardless of what the MBXVER program has determined, you are allowed to do that using the syntax:

**MBXVER** *MBX\_filespec* /V:3.0|4.0|4.1|4.5|5.0|5.5|6.0|6.5|7.0

That is, you can specify any one of the five versions 3.0, 4.0, 4.1, 4.5, 5.0, 5.5, 6.0, 6.5, or 7.0 as the required version, and it will be set into the MBX file. *Warning: You do this at your own peril!*

It is also possible to learn which are the keywords and functions used in a program that require a version of MapInfo beyond some version that you might have ... if, in fact, there are any such words in the program. To do this, you can invoke MBXVER in the following way:

**MBXVER** *MBX\_filespec* /W:3.0|4.0|4.1|4.5|5.0|5.5|6.0|6.5|7.0

The /W indicates that the words in the program beyond a certain version are to be listed. The version number after the /W: specifies that only words *beyond that version* are to be listed. If you specify /W without specifying a version, you will get all words beyond version 3.0.

It is possible to specify both the /V and /W parameters in the same command in order both to set the version and list the words. It doesn't matter in which order the parameters /V and /W appear.

If you ever forget the syntax for MBXVER, from a DOS command line, enter the following:

**MBXVER /?**

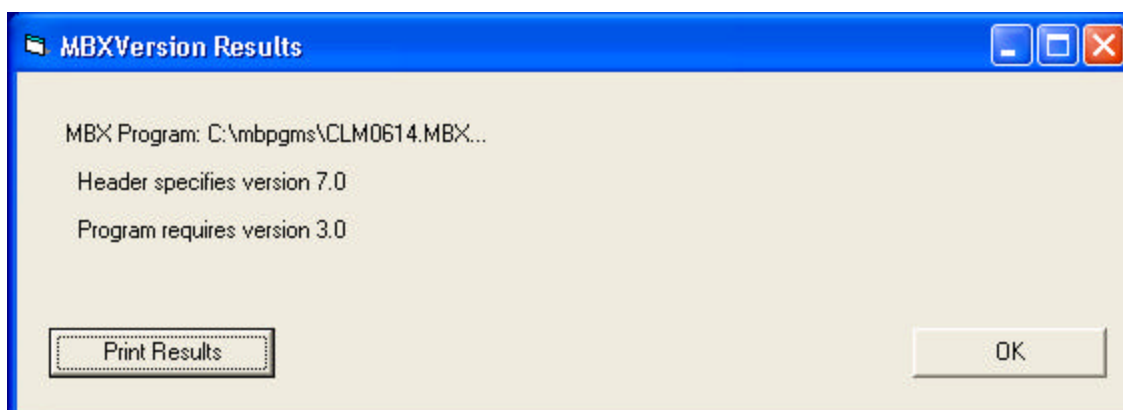
This provides you with help text for the syntax of the command. The help text is also produced if you invoke MBXVER without *any* command-line parameters.

### *The Windowed Program*

The program **MBXVERSION.EXE** is a Windows program that performs exactly the same functions as MBXVER.EXE, but does so in an interactive window. Invoke it using Windows **Start**, then **Run**, then specify **MBXVERSION** with a path indicating the folder in which the MBXVERSION program resides. The program presents the window:

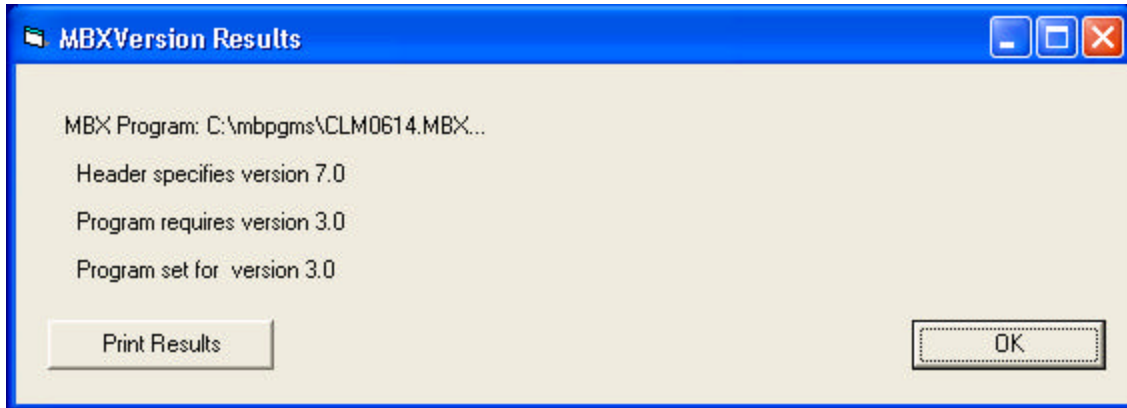


Select the MBX program file to be dealt with (you may browse using the button labeled "..."). If you do not check either checkbox below this, you simply get a *report* on the version currently set for the program, and the version that should actually be required ... for example:

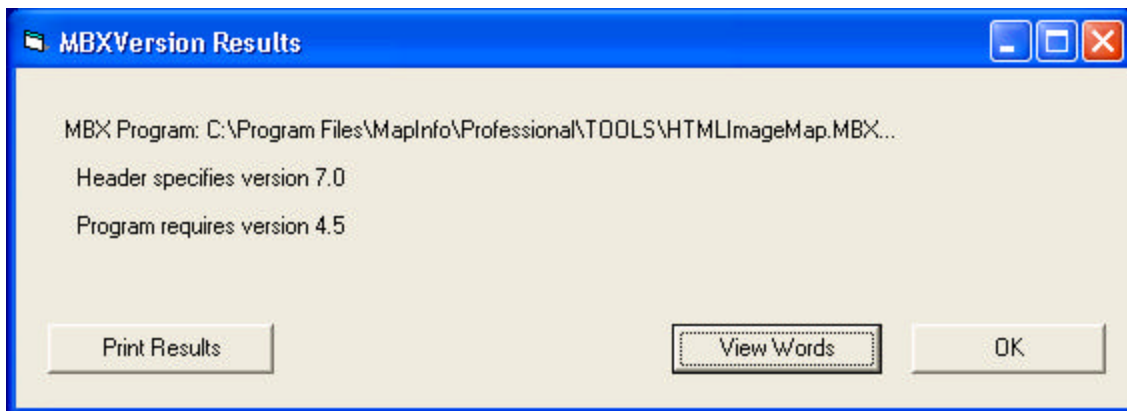


Clicking on the **Print Results** pushbutton allows you to print these results on paper.

To *set* the version of a program, you must check the **Set Version ...** check box. The default setting in the associated dropdown list box is **LowestValid**, indicating that MBXVERSION should set the version of this MBX program to the lowest version that is consistent with the facilities called upon by the program. By dropping down that list box, you can *force* the version to any version ... with the same caveat as we gave earlier. In any case, the program reports the version to which the MBX program was changed:

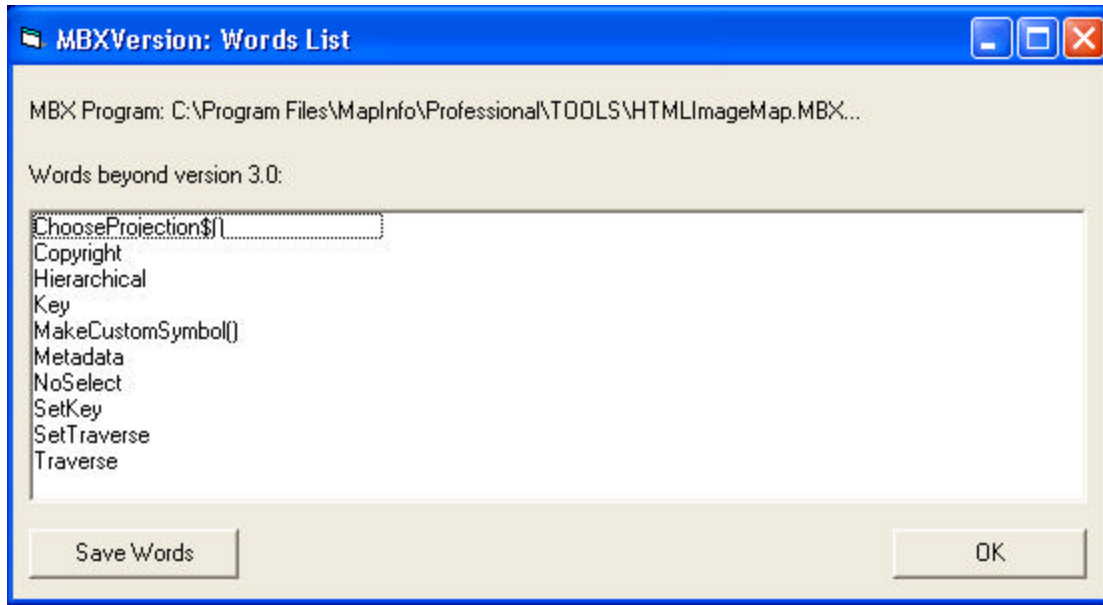


Finally, you can generate a list of keywords / functions used in the program that cause it to require a version *beyond* some lower version you may specify. To generate this list, check the **Report ...** check box, and specify the threshold version in the associated dropdown list box. You receive a report such as:



If you press the **View Words** pushbutton, you see a list of words (in this case, beyond Version 3.0):

---



If you wish, you can save this set of words to an ASCII text file by pressing the pushbutton **Save Words** and completing that dialogue.

### ***The Common DLL***

Both the command-line version and the windowed version call upon a DLL, **MBXVERFN.DLL**. It is this DLL which performs the analysis for each case. This DLL must reside in the same folder as the programs MBXVER.EXE and MBXVERSION.EXE.

### ***How It Works***

These programs understand the internal structure of a MapBasic executable (.MBX) program. For each statement type, each function, and each keyword in the MBX program, these programs determine the version of MapInfo required. They then take the *latest* of those versions as the *earliest* version of MapInfo that can run this MBX program. There are a few additional subtleties in the *manner* certain facilities are used that *could* require a later version than MBXVER/MBXVERSION detect, but experience suggests this will seldom be the case.

---

### ***Disclaimer and Usage Restrictions***

This complimentary program is provided, on an “as is” basis, as a convenience to our friends. Stopwatch Maps shall not be liable for any direct, indirect, consequential, or incidental damages arising from the use of this program.

This licensed program may be used freely by you for the purposes defined in this document. By receiving this program you agree not to sell it to others, and not to post it for download from a Web site or an FTP server. On the other hand, feel free to provide a link to our Web page from a Web page of your own.