

Installing and Running MusiX_{TEX} in MS Windows

Andre Van Ryckeghem, Don Simons

Contents

1	Introduction	1
2	File Locations; Unpacking the Distribution	2
3	Installing MusiX_{TEX} Files	3
3.1	Option A: Using a batch file	3
3.2	Option B: Manually transferring files	3
4	Running MusiX_{TEX} Manually	4
5	Running MusiX_{TEX} Using Batch Files	4

1 Introduction

This document explains in detail how to install and run MusiX_{TEX} under MS Windows XP, Vista, or 7. Along the way it includes information about locations of the various file types, which may be useful for other OS's or for any future incremental MusiX_{TEX} upgrades. Sections 2-3 provide the setup information in detail. Section 4 explains how to run MusiX_{TEX} manually, while Section 5 explains how to do it with batch files. For information about the MusiX_{TEX} language, see the [manual](#) included in `musixtex.zip`.

In order to use MusiX_{TEX} you must have installed other supporting software, including a $\text{T}_{\text{E}}\text{X}$ system, text editor, and a postscript viewer. This document assumes the $\text{T}_{\text{E}}\text{X}$ system MiK $\text{T}_{\text{E}}\text{X}$ 2.9 is already installed, but should apply at least to all versions since 2.6. The recommended postscript viewer is GSview (the executable is `gsview32.exe` or `gsview64.exe`). Ghostscript (executable `gswin32c.exe`) can produce PDF files from postscript. Some of the batch files in this distribution assume Ghostview or GSview is present.

Starting with a MusiX_{TEX} source file, a sequence of programs must be run to produce a PDF file of a music score. In practice these may be run in one of three ways: individually from a command line, with a batch file, or by using a shell. Section 4 lists the individual steps, and Section 5 describes the batch files provided with this distribution. Various shells are available but will not be discussed here.

MiK $\text{T}_{\text{E}}\text{X}$ itself comes with a version of MusiX_{TEX}. For various reasons it is suggested not to use these files. By installing the latest version of MusiX_{TEX} according to the guidelines given here, you will supersede the version supplied with MiK $\text{T}_{\text{E}}\text{X}$. It is also suggested that if MiK $\text{T}_{\text{E}}\text{X}$ ever offers to update your MusiX_{TEX} installation or provide a missing file, you politely decline, find the missing file yourself on the Icking archive, and position it in your own MusiX_{TEX} installation.

You may perform the installation either manually or by running the batch file `instmus.bat`. If you trust the batch file and are willing to accept default locations for the files, then after unpacking `musixtex.zip` you may in fact run the batch file right away, but we recommend you first read at least the next section, and note you'll still have to manually add a path to the PATH environment variable as explained there.

2 File Locations; Unpacking the Distribution

The paths to the default file folders are different in the different versions of Windows. They are listed in Table 1 below in terms of *environment variables*, along with the types of files that go in each folder. You can find out the true names for your Windows version by opening a command window (by running `cmd`), and then executing the command `set`.

Table 1. Folder names

Category	Description	Name
DOWNLOAD	Unzip <code>musixtex.zip</code> here	Arbitrary. Suggest <code>%USERPROFILE%\musixtex\musixtex115</code>
TOOLS	executables	Default: <code>%USERPROFILE%\Program Files\tools</code>
ROOT	all other MusiX _T E _X files	Default: <code>%APPDATA%\musixtexmf</code>
WORK	personal working files	Arbitrary

To begin installing MusiX_TE_X, create a DOWNLOAD folder, copy `musixtex115.zip` there, and unzip it, being sure to “use folder names.” After unzipping, the DOWNLOAD folder should have two subfolders, `InstallMusiXTEX` and `MUSIXTEXdistribution`, and a few text files.

Now, if you are willing to accept the default locations of Table 1 (and this is the recommended approach), you may skip the next step. Otherwise, to select your own locations, edit the batch file `SETMAPS.bat` in folder `%DOWNLOAD%\InstallMusiXTEX\InstallWindows` so that the appropriate `SET` commands reflect your choices. The folder names are entirely at your discretion (as long as you have write permission). Caution: any path name that contains spaces must be surrounded by double quotes when entered in the the batch file. It is possible but not recommended to put the files in the same folders with the corresponding MiK_TE_X files. The downside risk is that if MiK_TE_X is upgraded to a newer version, the MusiX_TE_X files will disappear.

The remainder of this section provides further detail about the various folders.

The TOOLS folder will hold various executables and batch files. Its name must manually be placed in the Windows path. One way to do this is to right click on the “My Computer” desktop icon, left click on Properties|Advanced|Environment Variables, in the “System Variables” section scroll down to “path”, select it, click edit, and append the full path name you have selected for the TOOLS folder. Note that if there are any command windows open at the time you do this, the path change will not affect them; it will only affect newly opened command windows. It may be tempting to avoid this path-setting exercise by using the folder that MiK_TE_X already established for its executables and placed in the path; however as noted above, this is not recommended because this folder will be overwritten every time MiK_TE_X is updated.

The ROOT folder will house all of the MusiX_TE_X files except the executables and your personal working files. MiK_TE_X needs to know where these files are. To accomplish that, the MiK_TE_X “Settings” program must be run. If you install using `instmus.bat`, this program will be started for you; otherwise you must run it manually from the Windows Start menu. After the program starts, select the “Roots” tab, click the “Add” button, and enter the full path name for your MusiX_TE_X ROOT folder.

A final folder, WORK, houses user-created MusiX_TE_X source files. It need not ever be formally specified, and can change at will. It is simply the place you keep your current working files. The logical place for this would be in a subfolder of your “My Documents” folder.

3 Installing MusiX_TE_X Files

3.1 Option A: Using a batch file

The batch file `INSTMUS.bat` in `%DOWNLOAD%\InstallMusiXTeX\installWindows` will copy all files to the proper locations, edit some configuration files, and launch the MiK_TE_X settings program. It will create a file `instmus.log` with a record of the installation. Run it either from the command line (after navigating to the folder that contains it) or by double-clicking it in Windows Explorer. When it launches the MiK_TE_X settings program, enter the ROOT path under the “Roots” tab, and refresh the filename database (FNDB).

3.2 Option B: Manually transferring files

Instead of using the batch file, you could perform all the steps manually. All the necessary steps are enumerated below.

1. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\fonts\mf*.mf` into the folder `%ROOT%\fonts\source\public\musixtex`
2. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\fonts\tfm*.tfm` into the folder `%ROOT%\fonts\tfm\public\musixtex`
3. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\tex*.*` into the folder `%ROOT%\tex\generic\musixtex`
4. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\musixtexadd*.tex` into the folder `%ROOT%\tex\generic\musixtex`
5. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\bin*.*` into the TOOLS folder.
6. Copy all the files `%DOWNLOAD%\MUSIXTEXdistribution\fonts\type1*.pfb` into the folder `%ROOT%\fonts\type1\public\musixps\type1`
7. Copy the pdf configuration file `c:\Program Files\MiKTeX 2.9\dvipdfm\config\configto` `%ROOT%\dvipdfm\config\config`. Edit the copied file by adding the line
`f musix.map`
8. Copy the file `%DOWNLOAD%\MUSIXTEXdistribution\dvipdfm\musix.map` to `%ROOT%\dvipdfm\config\musix.map`. Edit the copied file by adding the line
`f musix.map`
9. Copy the postscript configuration file `c:\Program Files\MiKTeX 2.9\dvips\config\config.ps` to `%ROOT%\dvips\config\config.ps`. Edit the copied file by adding the line
`p +musix.map`
10. Copy the file `%DOWNLOAD%\MUSIXTEXdistribution\dvips\musix.map` to `%ROOT%\dvips\config\musix.map`
11. Copy the file `%DOWNLOAD%\MUSIXTEXdistribution\dvips\psslurs.pro` to `%ROOT%\dvips\base\psslurs.pro`
12. Copy the file `c:\Program Files\MiKTeX 2.9\dvips\tetex\config.pdf` to `%ROOT%\dvips\tetex\config.pdf`. Edit the copied file by adding the line
`p +musix.map`

- (Optional) If you wish to use the provided batch files that refer to `gsview.exe` or `gswin32c.exe`, or if you wish to run these programs from a command line without typing the full path, you should locate `gswin32c.exe` and the `gsview` executable (`gsview32.exe` or `gsview64.exe`), copy them into the `TOOLS` folder, and rename the one for `gsview` to `gsview.exe`.

Run the MiKTeX “settings” program from the “start” menu and click the button labelled “Refresh FNDB” (File Name DataBase).

ALWAYS REMEMBER TO REFRESH THE FILE NAME DATABASE whenever you add a file to MusiXTeX.

4 Running MusiXTeX Manually

As already mentioned, to produce a score you may run the necessary sequence of programs manually from a command line (and some from the Windows GUI), by using a batch file, or by using a shell. This section explains how to do it manually, while the next one explains how to use the batch files that are supplied with this distribution.

To test your setup, apply the steps outlined below to any legitimate MusiXTeX file, for example the provided file `%DOWNLOAD%\InstallMusiXTeX\TestMusetup\test1.tex`. The instructions below will in fact assume this is the file being processed.

Begin by moving or copying `test1.tex` to your selected working folder. This can be anywhere you have write permission. Open a command window (Start|Run, then enter `cmd`) and navigate to the working folder.

MusiXTeX itself is a 3-pass system. For the first pass, enter `etex test1`. This should produce `test1.mx1`. Next, enter `musixflx test1`. This should produce `test1.mx2`. Finally, enter `etex test1`. This should produce `test1.dvi`. The basic processing is now complete.

The files `test1.mx1` and `test1.mx2` are intermediate working files produced by MusiXTeX (on the first pass) and `musixflx` respectively. Once the 3-step process is complete, they will never be needed again and should be deleted; in fact, if you have made any changes to your MusiXTeX file that affect horizontal spacing, you *must* delete `test1.mx1` before repeating the 3-step process. A great deal more information about the process is included in the MusiXTeX [manual](#) in `%DOWNLOAD%\MUSIXTEXdistribution\doc\musixtex\manual\musixdoc.pdf`.

To view the result, you should create a postscript file by typing `dvips test1`. This should produce `test1.ps`, which can be viewed by running `GSview` from the Windows GUI. Note that all TeX systems include a “DVI viewer” such as `YAP`. These will not display Type K postscript slurs, and so are generally not recommended.

If you want to produce a PDF file from the postscript, you can do it from within `GSview`, by going to `File|Convert` and selecting device “`pdfwrite`”.

5 Running MusiXTeX Using Batch Files

This distribution includes a set of batch files that run MusiXTeX and perform other associated functions. `instmus.bat` copies them all into the `TOOLS` folder, so they will be in the path. Most users who don’t use a shell will ultimately want to use batch files, especially if they use one or both of the preprocessors `M-Tx` or `PMX`.

The programs called by the batch files include `gsview.exe`, `gswin32c.exe`, `etex.exe`, `latex.exe` and `musixflx.exe`. Table 2 defines abbreviations for these programs. The abbreviations are com-

bined to create the names of the batch files as given in Table 3. For example, `mupsall.bat` means: For all MusiX_TE_X files in the current folder, make a Postscript file.

Table 2. Abbreviations used in batch file names

MusiX _T E _X	La _T E _X	Postscript	PDF	PS or PDF viewer	All files in folder
<code>mu</code>	<code>la</code>	<code>ps</code>	<code>pdf</code>	<code>view</code>	<code>all</code>

Table 3. Batch files for running MusiX_TE_X

Batch file	Produces:	Comment
<code>mups.bat</code>	ps-file	MusiX _T E _X source → E _T E _X → <code>musixflex</code> → E _T E _X → dvi → <code>dvips</code> → ps
<code>mupsview.bat</code>	ps-file	MusiX _T E _X source → <code>mups.bat</code> → viewer(PS)
<code>mupdf.bat</code>	ps-file, pdf-file	MusiX _T E _X source → <code>mups.bat</code> → <code>gswin32c</code> → pdf
<code>mupdfview.bat</code>	ps-file, pdf-file	MusiX _T E _X source → <code>mupdf.bat</code> → viewer(PDF)
<code>laps.bat</code>	ps-file	La _T E _X source → La _T E _X → <code>musixflex</code> → La _T E _X → dvi → ps
<code>lapsview.bat</code>	ps-file	La _T E _X source → <code>laps.bat</code> → viewer(PS)
<code>lapdf.bat</code>	ps-file, pdf-file	La _T E _X source → <code>laps.bat</code> → <code>gswin32c</code> → pdf
<code>lapdfview.bat</code>	ps-file, pdf-file	La _T E _X source → <code>lapdf.bat</code> → viewer(PDF)
<code>mupsall.bat</code>	ps-file, no view	processes all MusiX _T E _X files in current folder with <code>mups.bat</code>
<code>mupsviewall.bat</code>	ps-file	processes all MusiX _T E _X files in current folder with <code>mups.bat</code> and display PS
<code>mupdfall.bat</code>	ps-file, pdf-file, no view	processes all La _T E _X files in current folder with <code>mupdf.bat</code>
<code>mupdfviewall.bat</code>	ps-file, pdf-file	processes all MusiX _T E _X files in current folder with <code>mupdf.bat</code> and display PDF
<code>lapsall.bat</code>	ps-file, no view	processes all La _T E _X files in current folder with <code>laps.bat</code>
<code>lapsviewall.bat</code>	ps-file	processes all MusiX _T E _X files in current folder with <code>laps.bat</code> and display PS
<code>lapdfall.bat</code>	ps-file, pdf-file, no view	processes all La _T E _X files in current folder with <code>lapdf.bat</code>
<code>lapdfviewall.bat</code>	ps-file, pdf-file	processes all MusiX _T E _X files in current folder with <code>lapdf.bat</code> and display PDF
<code>mymp.bat</code>	all	processes <code>mymp.tex</code> – make your own batchfile with this example

To use any one of the first eight batch files (without “all” in the name), you must navigate to your working folder (containing your `.tex` file), then run the batch file from the command line and supply it with one argument, namely the basename (the name without any extension) of the file

you want to process. For example, to make and view a postscript from `myp.tex`, you would type `mupsvie w myp` on the command line. For many users this may be an entirely adequate method. It would allow you to have more than one different `.tex` file in the working folder.

On the other hand, the batch files `*all.bat` do not require any arguments, but they will process *all* of the `.tex` files in the folder from which they are called. Since they are in the path, they can be invoked from command line in the working folder. Alternatively, you could make a copy of the desired `*all.bat` in the working folder, and then run it by clicking on it from Windows Explorer. Obviously these batch files can't be used if there is more than one `.tex` file in the folder, unless of course you want more than one file to be processed.

Finally, there is a third way to process with a batch file whose only disadvantage is that you must create a very simple, new, single-line batch file for each basename. The new batch file does not require any arguments and may therefore be run by clicking on it from Windows Explorer, and it will only process the one desired file. For example, to create and view a postscript from `myp.tex`, create and run a batch file named `myp.bat` with the single line `call mupsvie w.bat %~n0`. This arcane syntax takes the basename of the batch file itself and passes it as an argument to the called batch file.