

Installing and Running the Stand-alone PMX Distribution

Version 3.00 – February 2023

Don SIMONS
dsimons@roadrunner.com

1 Introduction

This document explains in detail how to install and run **PMX** in a Windows operating system, using the "stand-alone" distribution [pmx300.zip](#), which resides in the [Werner Icking Music Archive](#) (WIMA). That is the basic distribution assembled by the code's author. The executable files in it will only run under Windows. For other operating systems, other options include either compiling the FORTRAN source files in `pmx300.zip` and then following or adapting the remainder of the guidance provided here, or (for Windows as well) using other distributions which will be derived in due course from `pmx300.zip`, archived in various T_EX repositories, and supplied with their own installation guides. Links to those other repositories can be found in the [software section](#) of the Werner Icking Music Archive (WIMA).

It is assumed here that T_EX and MusiX_TE_X version 1.21 or later are already present. If not, WIMA also provides instructions and links for acquiring and installing those programs from the same repositories.

2 Setup from scratch

Assuming T_EX and version 1.21 or later of MusiX_TE_X have already been installed, decompress `pmx300.zip` to any temporary folder. It will contain the FORTRAN source files `pmxab.for` and `scor2prt.for`, Windows executables `pmxab.exe` and `scor2prt.exe`, several sample `.pmx` files, the full manual `pmx300.pdf`, a very condensed summary of all **PMX** commands `ref300.pdf`, and this installation manual `install_run_PMX300.pdf`. If necessary, compile the FORTRAN programs. I have tried to keep the source code as generic as possible, but minor modifications may be needed for FORTRAN-to-C translation and/or other compilers.

Once you have assembled a full set of files, put the executables somewhere in your search path or in your working directory, `pmx.tex` somewhere where T_EX can find it (which may or may not be part of a TDS-compliant file structure), and the sample `*.pmx` files in your working directory (the one from which you will run **PMX**). The rest of the files can reside wherever is convenient. Refresh the T_EX file name database.

3 Updating an existing PMX installation

If you already have an older version of **PMX** installed, then after unzipping the distribution to a temporary folder, simply replace the old copies of `pmxab.exe`, `scor2prt.exe`, and `pmx.tex`. Nothing more needs to be done, except to ensure that you have version 1.21 or later of MusiX_TE_X installed.

4 Running PMX

Running **PMX** will be demonstrated by example, with the file `barsant.pmx`. The first thing you need to do is specify where the generated \TeX file is to be placed. The path to that location is part of the setup data in the **PMX** source file, in this case the 15th line of `barsant.pmx`. It is set to `./` which represents the current folder, the one from which you will run **PMX** and the one where `barsant.pmx` resides. If for some reason you want the \TeX file to go somewhere else, then replace `./` with a path—either relative or absolute—to the desired folder.

If you haven't done so, open a command window and navigate to the folder containing `barsant.pmx`. Execute **PMX** by typing `pmxab barsant .`. Alternatively, you may just type `pmxab` <return> and you will be prompted for a jobname, which in this case is just `barsant . pmxab` will generate two files in the working directory: `barsant.pml` is a log file, and `pmxaerr.dat` contains a single integer, 0 if the run was successful, otherwise the line number in the `.pmx` file of the fatal error (useful for batch processing). Also, on successful completion, `barsant.tex` will be placed in the path specified in the setup.

Now you are right where you would be after entering, debugging, and rough-editing the `.tex` file manually. To see the results, process `barsant.tex` just as you would for any \MusixTeX file. You must first run `etex`, then `musixflx`, and then `etex` again. You may view the `.dvi` file so produced with a program like `yap`, or go on and run `dvips` to create a postscript file and view that with a postscript viewer such as `GSview`. If you have used postscript slurs, dvi viewers have the disadvantage that they will not display them.

To make separate parts, run `scor2prt` by typing `scor2prt barsant`. The program will create a new `.pmx` file for each instrument, in this case `barsant1.pmx` and `barsant2.pmx`. Then to view and print the individual parts, you may process the two `.pmx` files like you did the one for the score.

5 Using scripts to ease processing

Rather than having to manually run five programs every time you want to process and view a `.pmx` file, you may use a batch script. Such scripts are generally platform-dependent.

For Windows, the following is a possible batch script which produces a postscript file:

```
pmxab %1
echo off
echo errorlevel
echo %errorlevel%
if not errorlevel 1 goto continue
echo stopping
goto :end
:continue
del *.mx?
etex %1
musixflx %1
etex %1
dvips -00.25in,-0.22in %1
:end
```

To use it, copy the script into a text file, save it with a name like `gopmx.bat`, and execute it by typing `gopmx barsant`.

At the `dvips` stage, this script uses offsets appropriate to the default output of **PMX** with letter-size paper, assuming that the **PMX** option `Acl` has *not* been invoked.

If you wish to go on and produce a PDF file, most T_EX systems come with a program `ps2pdf`. You could run this manually or include it in the main batch script. It may require that the file name provided to it include the extension `.ps`. Alternatively, the windows postscript viewer `GSview` has an option `convert` for producing PDFs.

If you have installed **PMX** other than as described here, the distribution you used may include its own batch script and instructions for end-to-end processing.