# **Installation and Configuration Instructions for MPSS, Inc., Open Source Components**

October 12, 2018

As part of a presentation by MPSS, Inc., owner Jim Heuer at the 2018 SW Fox Conference, MPSS, Inc., is releasing a number of Python modules of interest to VFP developers into Open Source. All of these modules are released under the GNU Lesser GPL, Version 3.0, which allows use of the software either in other Open Source projects or in commercial products. This document is copyrighted by MPSS, Inc., and also released into Open Source under the GNU Lesser GPL, Version 3.0.

As of this version 0.5 release, no formal Python install routines are provided, so these files must be manually installed for use with your Python applications. It is hoped that by early 2019, we will have released versions of these modules in a form installable by pip or easyinstall utilities.

**General Installation Instructions**

Be sure to read ALL of this information before attempting installation, especially if you are not familiar with Python module installations.

By default, 3rd party libraries are installed in the site-packages directory of your Python installation. If you are using Python 3.7, this will likely be in:

C:\Python37\Lib\site-packages

Other Python versions will typically be in a directory off the root of C: with the version indicated in the directory name (e.g. c:\Python36). (Some Windows installers will put the Python37 directory under one of the Microsoft-defined directories. In that case, we recommend that you install Python v.3.7 into C:\Python37 instead.)

For manual installation of all .py, and .pyd files, copy them directly into the site-packages directory.

**CodeBase Tools**

If you are installing the CodeBaseTools module, it requires the CodeBasePYWrapper.pyd file. Three versions of the .pyd file are provided, one for Python 2.7, one for Python 3.6, and one for Python 3.7. These are not interchangeable. There is no plan at this time for a Python 2.6 version, but if there is a demand for it, I’ll create one.

Copy the CodeBaseTools.py file and the appropriate CodeBaseBYWrapper.pyd file into the directory as described above. You will also need MPSSBaseTools.py copied into that same directory. The .py files are designed to be cross compatible between Python 2.7 and Python 3.x versions.

The source code for the CodeBasePYWrapper.C program is provided, but is not needed unless you decide to make modifications and recompile it. For guidance on recompilation, contact MPSS, Inc.

CodeBase Tools provide a wrapper to simplify use of the CodeBase library of C functions in Python. The file c4dll.dll is required to be available to the Python interpreter as well as the zlib.dll. Both of these executables are available as Open Source under the GNU Lesser GPL, V3.0 license. They are found in the CodeBaseEngine directory of this distribution and should be copied into the main Python directory (normally c:\python27 or c:\python36, etc. see general instructions).

CodeBase tools are available only for 32-bit Python at this time. But only 32-bit Python provides COM-based interoperability with Visual FoxPro, so this may not be a limitation for you.

**MPSSBaseTools**

This library of useful utilities, a number of which are direct replacements for functions or commands in VFP with no direct equivalent in Python, may simply be copied to the site-packages directory as explained in General Installation Instructions.

**Excel Tools and the DBFXLStools2 Module**

These two components make for quick and easy transformation of Excel spreadsheets into DBF tables, and vice versa. They require the LIBXL product, which may be purchased from libxl.com for a reasonable price. The libxl.dll file, once acquired, should be copied to the main Python install directory (see above for c4dll.dll instructions).

If you just want the Excel Tools without the DBF conversions, you need to copy only the ExcelTools.py and the LibXLWrapper.dll into the site-packages directory per the general instructions. The .dll files for these components are usable under all 32-bit versions of Python, and ExcelTools.py as of this writing has been fully tested with Python 3.6 and Python 3.7.

The producers of LIBXL will supply you with a user name and a license key string once you have purchased their product. (You can download the .DLL and use it in test mode without the purchased key, but that will result in LibXL copyright notices embedded in your spreadsheets and other limits on the number of rows you can process.) These two values should be pasted into the LibXLLicenseInfo.TXT file: the user name on the first line and the license key on the second. This file should be placed in the same directory as the LibXLWrapper.dll file, and is required for proper functioning of the components.

If you want the conversion capabilities of DBFXLStools2, you should copy that .py file into the site packages too. You must install the ExcelTools module for DBFXLStools2 to function. This includes purchasing a licensed version of the LibXL product.

**ExcelComTools**

To use the ExcelTools module directly as a COM server for a VFP application separately from the DBF to XLS conversion capability (e.g. you want to create XLS or XLSX files under program control from your VFP application), you can use the ExcelComTools COM object. This requires the pycomsupport.py module, which is also provided. This module supports both v2.7 and v3.x versions of Python. If using a 3.x version of Python and you find the COM server cannot be accessed by your VFP apps after you have registered the server, see the discussion of this issue in the White Paper for the 2018 SW Fox Session.

**Using the Modules**

As of this writing, the available documentation is found in the “doc strings” contained in the .py modules. This information is extensive, and we hope it will be sufficient to get you started. The Session Examples ZIP file contain examples of the use of some of these tools, and should also provide you with some further information.

For more readable and accessible documentation on the functioning of these modules, see the Documentation directory where you’ll find a file named “MPSSCommon Module Docs.html” which is the home HTML file for an indexed, formatted, searchable version of the doc strings in all the modules. This was produced by the PyPI module called sphinx.