

Installing Volatility Plugins (Windows)

So you've already installed [Volatility using SVN](#) and you want to try out some of the community plugins that people are raving about. Publicly known plugins are listed on the [forensics wiki](#). The wiki contains links to plugins as well as links to blogposts/articles for further information on installation, dependencies and how they work.

Most plugin installation is straightforward where one may simply place the plugin in the **memory_plugins** directory within the Volatility directory. Some are only slightly more complicated by needing a helper library installed in addition to the plugin itself. Others are even more complicated and require some installation of Python libraries which may or may not need the help of other compiled libraries. Therefore we have three cases for plugin installation (please visit the forensics wiki for more information):

1. Simple Case - only in **memory_plugins**
 - volshell
 - IDT
 - cryptoscan
 - orphan_threads
 - keyboardbuffer
 - getsids
 - moddump
 - objtypescan
 - symlinkobjscan
 - driverscan
 - fileobjscan
 - pstree
2. More Complex Case - also supporting file(s)
 - driverirp (needs driverscan)
 - threadqueues (needs [lists.py](#))
 - ssdt (needs [lists.py](#))
3. Most Complex Case - installation of supporting libraries
 - malfind (needs [pydasm](#) and [pefile](#))
 - kernel_hooks (needs [pefile](#))
 - usermode_hooks (needs [pefile](#))
 - volreg (needs [pycrypto](#))
 - VolRip (needs volreg and [Inline::Python](#))

Installing Volatility Plugins (Windows)

Simple installation of volshell

For an example of a simple installation, we will install the [volshell](#) plugin. Simply download the [volshell.py](#) file and place it into your **memory_plugins** directory. You can test to make sure that is installed correctly by running Volatility without any arguments and volshell should appear under "Supported Plugin Commands" highlighted below in Figure 1. All other "simple case" plugins should install the same way.

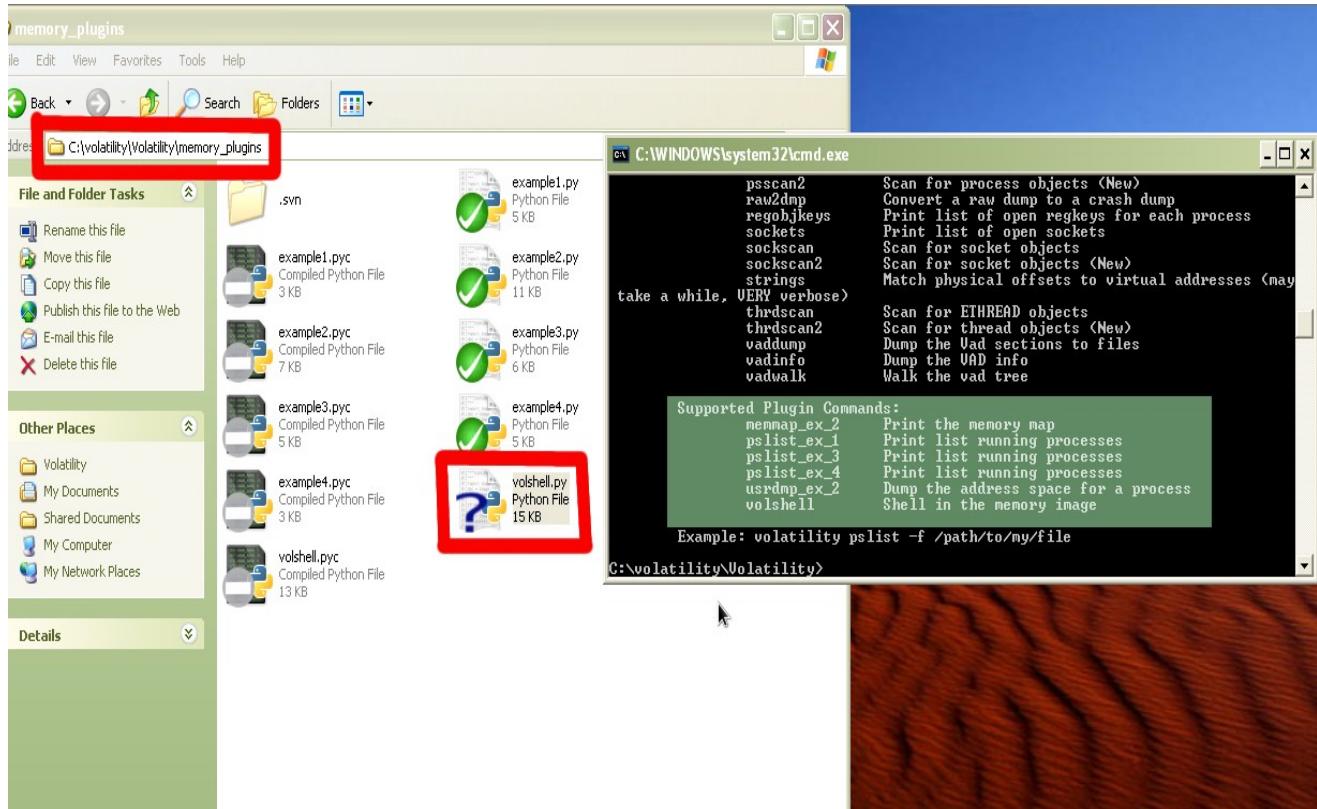


Figure 1: Installation of volshell

More Complex Cases

The [ssdt](#) and [threadqueues](#) plugins require that the [lists.py](#) library file be placed in the **forensics/win32** directory in addition to placing the [ssdt.py](#) and [threadqueues.py](#) into the **memory_plugins** folder as before. The [driverirp](#) plugin requires the [driverscan](#) plugin in order to work. Both of these plugins are placed in the **memory_plugins** directory.

After placing the files in the appropriate places, check to make sure that the plugins are properly installed by running volatility without any arguments as before and checking under "Supported Plugin Commands" (Figure 1).

Installing Volatility Plugins (Windows)

Most Complex Cases

For the "most complex cases" other libraries must be installed for the plugin to work properly. First we will look at installing the [malfind](#) plugin. First of all, download the [malfind.py](#) plugin file and place it in the **memory_plugins** directory. Now you must install the [pydasm](#) and [pefile](#) libraries.

In order to install the pydasm library, you will have to do some initial setup including by installing a gcc compiler and make. For this tutorial, we will use [MinGW](#).

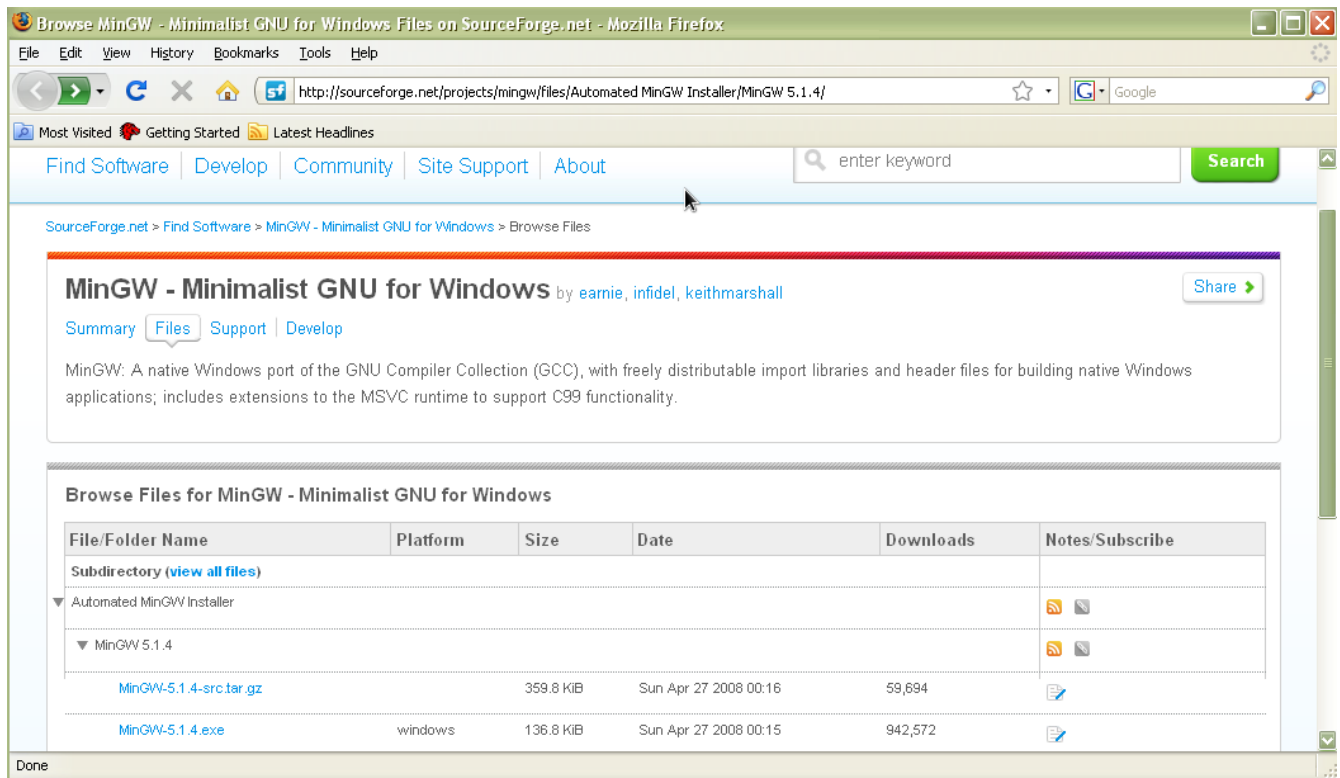


Figure 2: Sourceforge download site for MinGW

Download the windows installer for MinGW from the sourceforge website (Figure 2). Double click to install (Figure 3-9).

Installing Volatility Plugins (Windows)



Figure 3: Choose "Download and Install"



Figure 4: Click "Agree"

Installing Volatility Plugins (Windows)



Figure 5: Choose "Current"

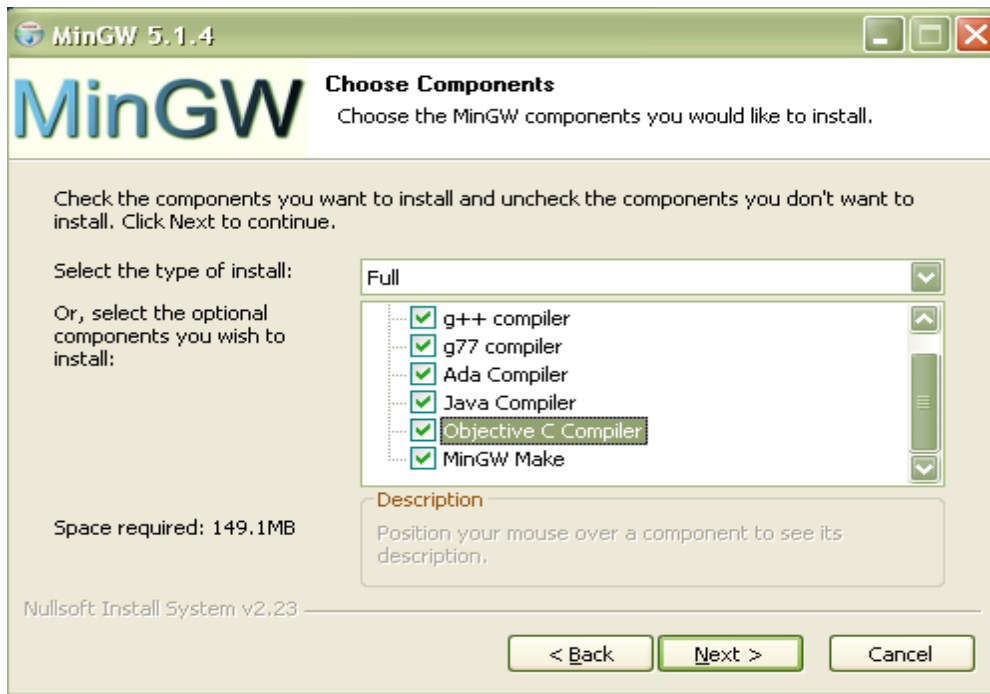


Figure 6: Choose compilers and MinGW make

Installing Volatility Plugins (Windows)

You do not necessarily have to install all compilers however, for simplicity, do a full install.

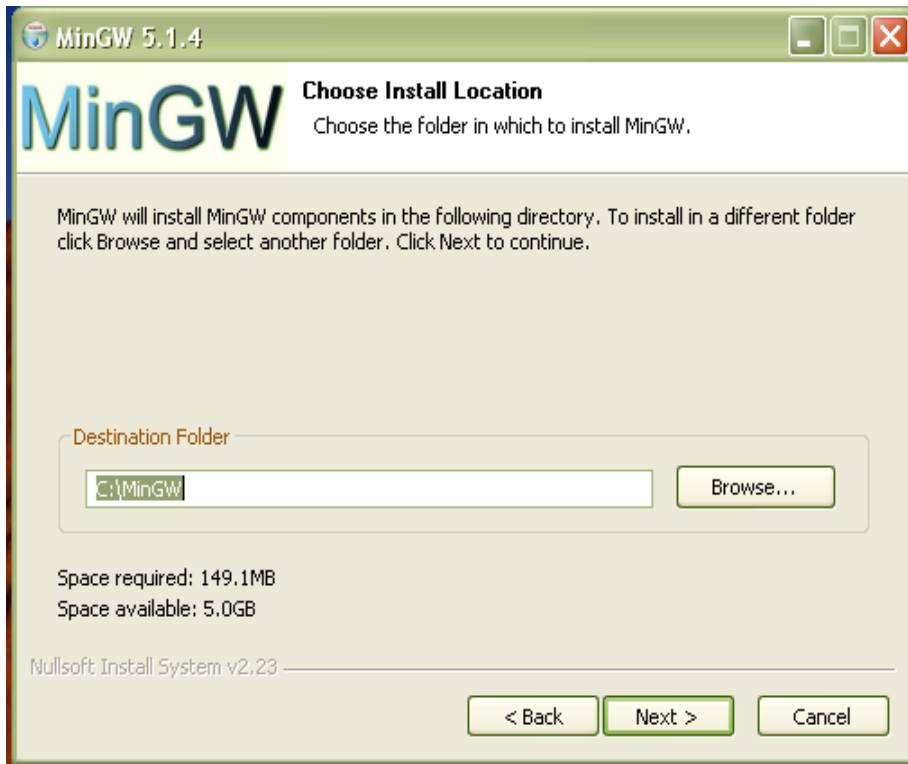


Figure 7: Choose location for installation. The default is fine.

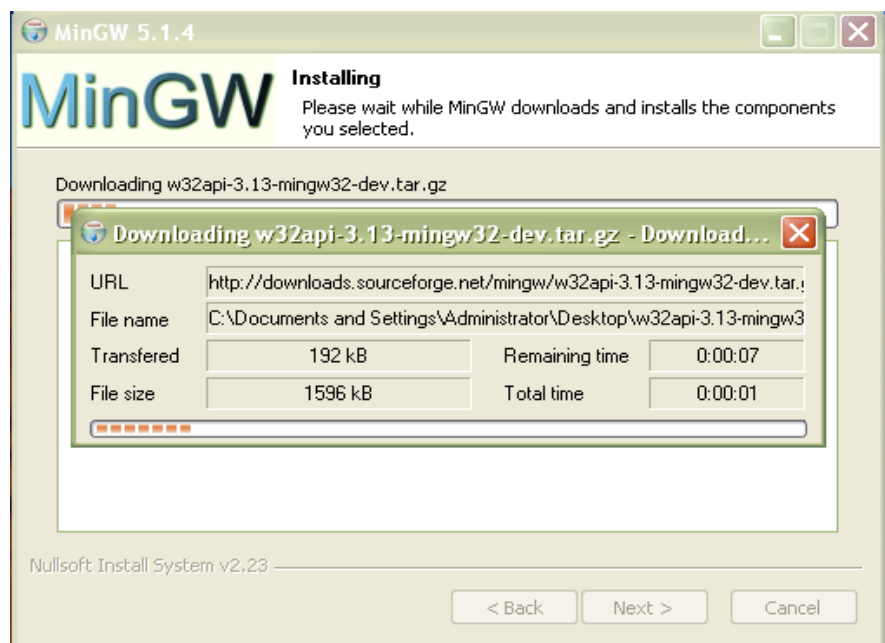


Figure 8: Installing MinGW

Installing Volatility Plugins (Windows)

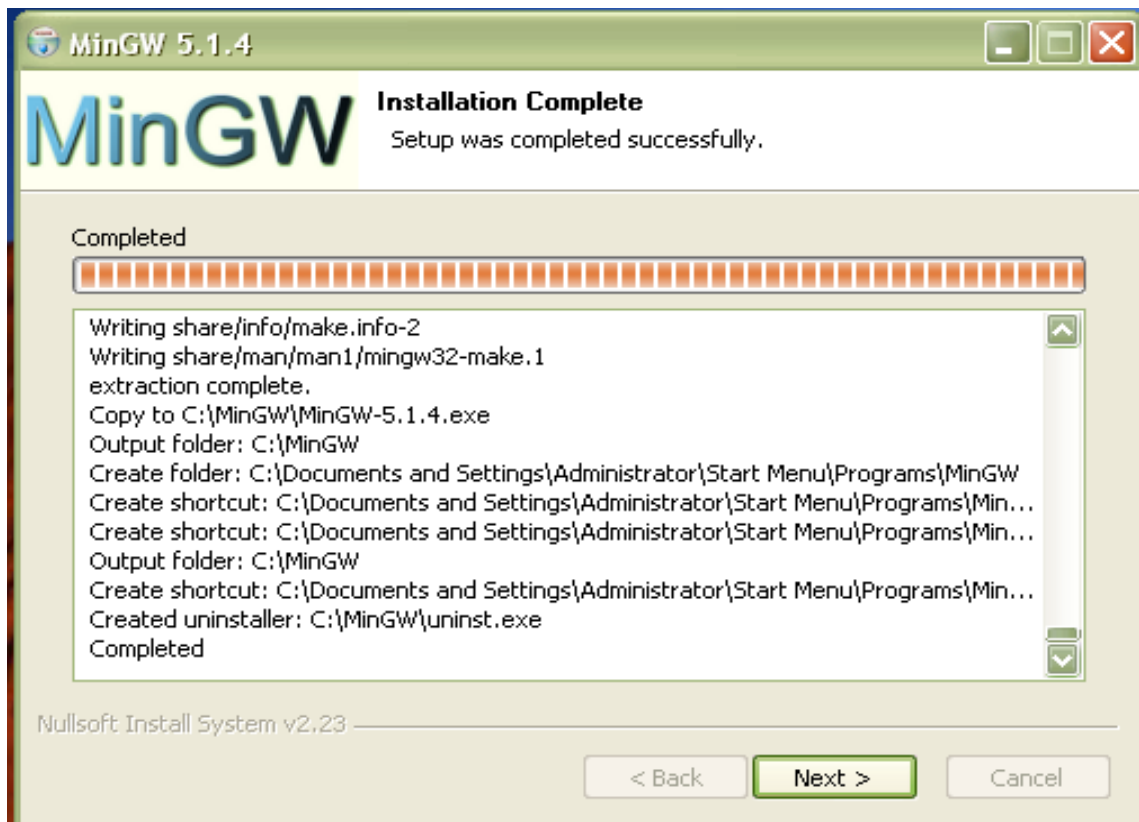


Figure 9: installation complete

Once the installation is complete and you have clicked finish, you will have to make a few adjustments to the installation in order to get things working properly. First of all, we need to have an executable called "make.exe". The **make** executable for MinGW is appropriately named **mingwmake.exe**. Simply copy this executable and paste it into the same directory (C:\MinGW\bin) which should result in an identical copy named "Copy of mingwmake.exe". Rename this executable to "make.exe" as shown in Figures 10-11.

Installing Volatility Plugins (Windows)

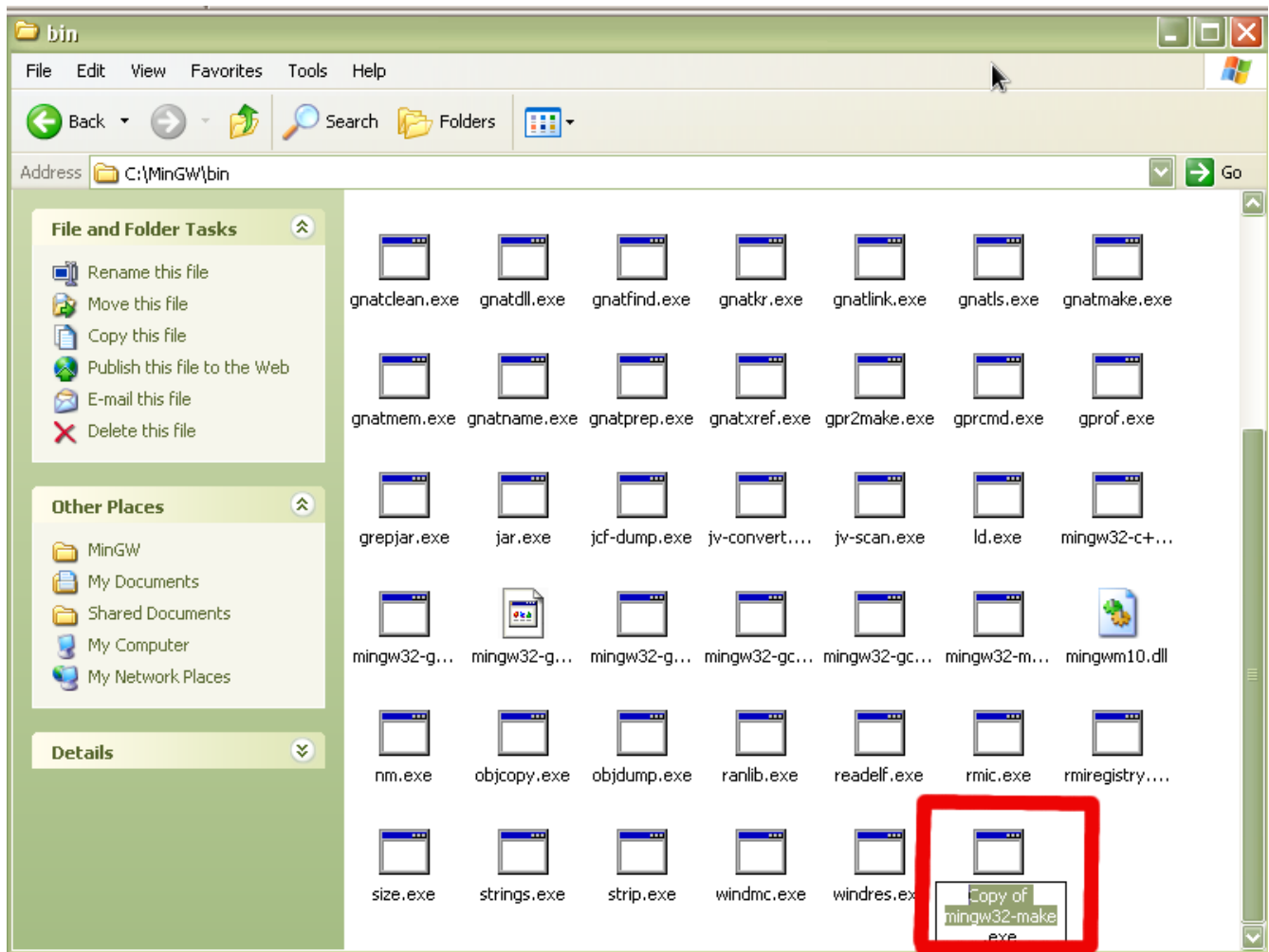


Figure 10: "Copy of mingwmake.exe"

Installing Volatility Plugins (Windows)

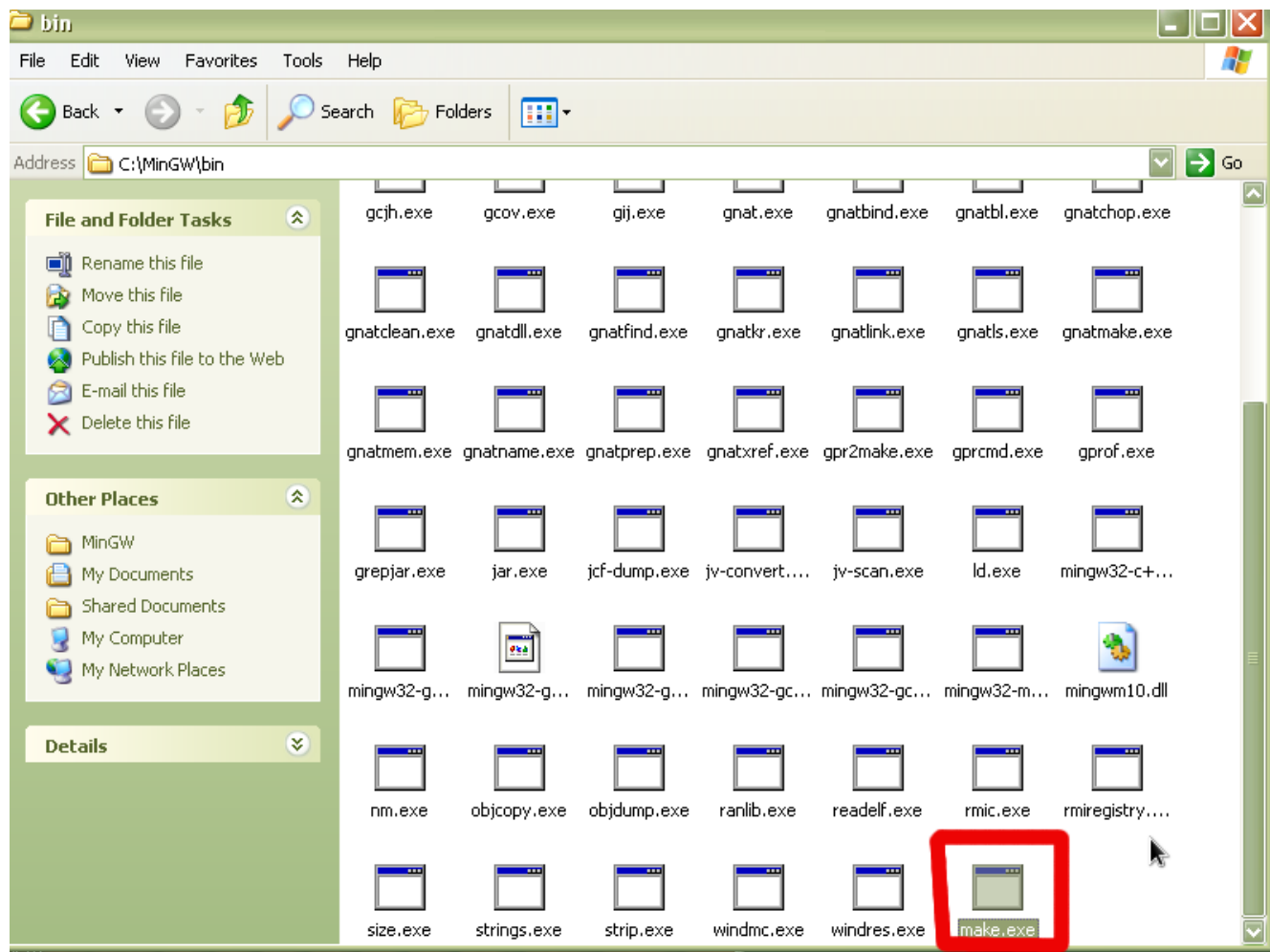


Figure 11: Rename to "make.exe"

Now we have to modify our path to include the executables for MinGW. If you have a regular start menu, click on start and then right click on "My Computer" and choose properties. If you have the classic start menu, just right click on "My Computer" and choose properties. Click on the "Advanced" tab and then click on "Environmental Variables". Click on the Path system variable towards the bottom of the window and click the "Edit" button. We will append the path of our Python installation to the end of the existing Path variable. Where it says "Variable Value" go to the end of the line and add the following (if you installed MinGW in a different location, modify appropriately):

;`C:\MinGW\bin`

Installing Volatility Plugins (Windows)

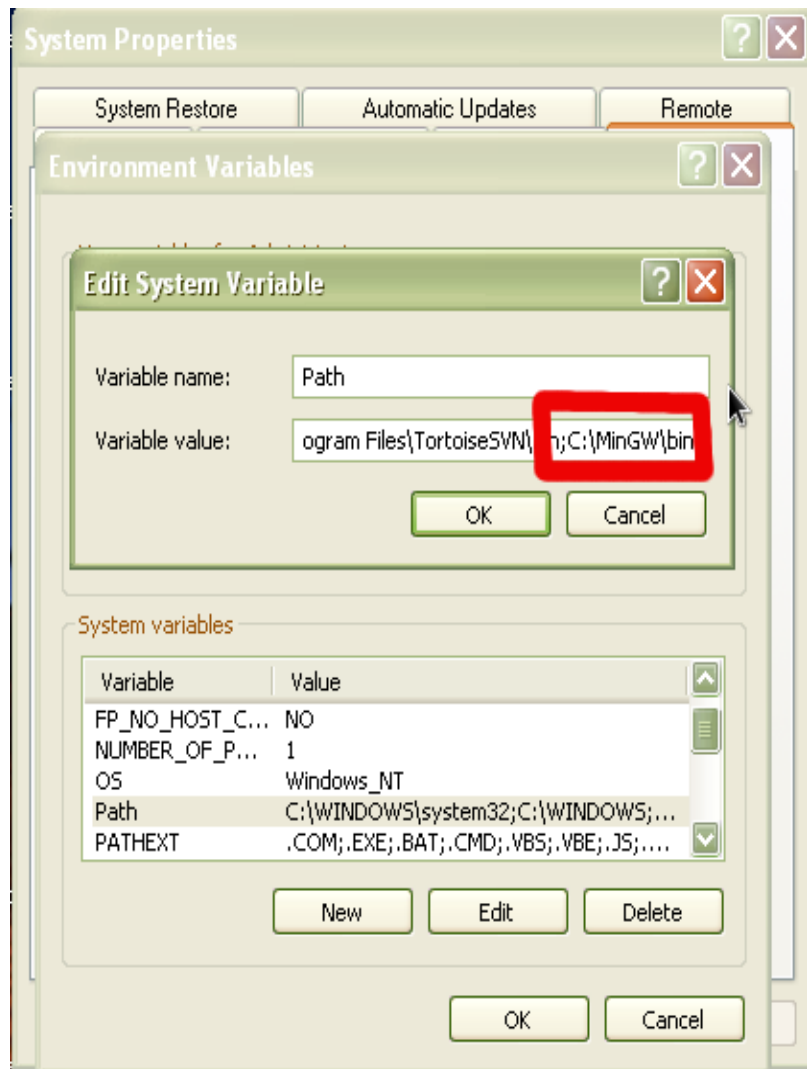
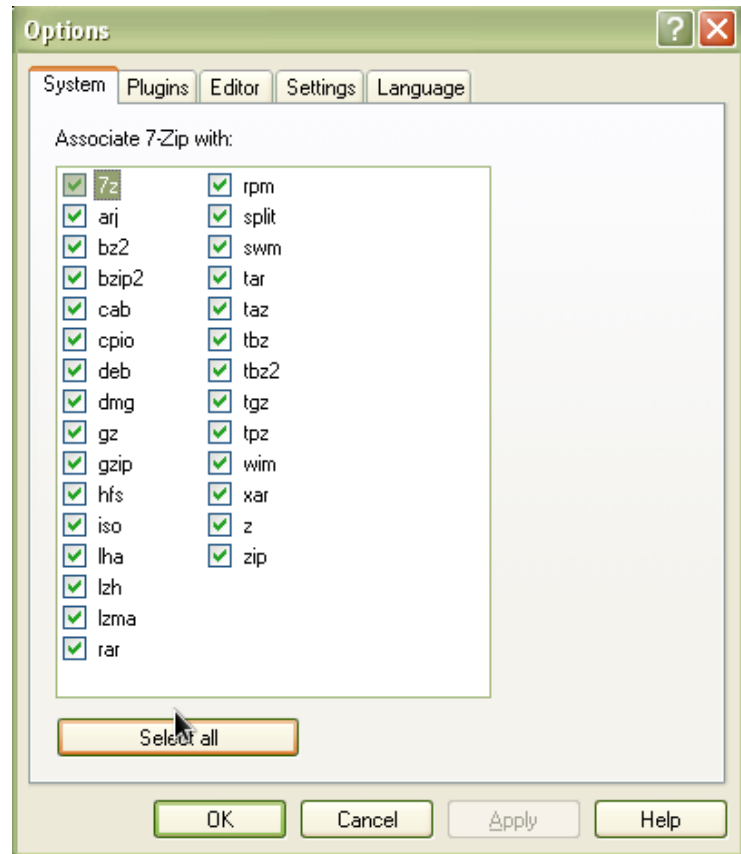


Figure 12: Adding C:\MinGW\bin to the path variable

Now for installing pydasm. Download the [source code for libdasm](#). The easiest way to extract the contents from this tar ball is using [7zip](#). Once you have 7zip installed, you can associate all zip files by starting the 7zip Filemanager (Start->Programs->7-zip->7-zip File Manager) and clicking on "Tools->Options" and clicking "Select all" in the system tab and "OK" (Figure 13).

Installing Volatility Plugins (Windows)

Figure 13: Associating zip file types



At this point you are ready to extract the libdasm/pydasm source code. Double click the downloaded pydasm tar ball. You should see the following:

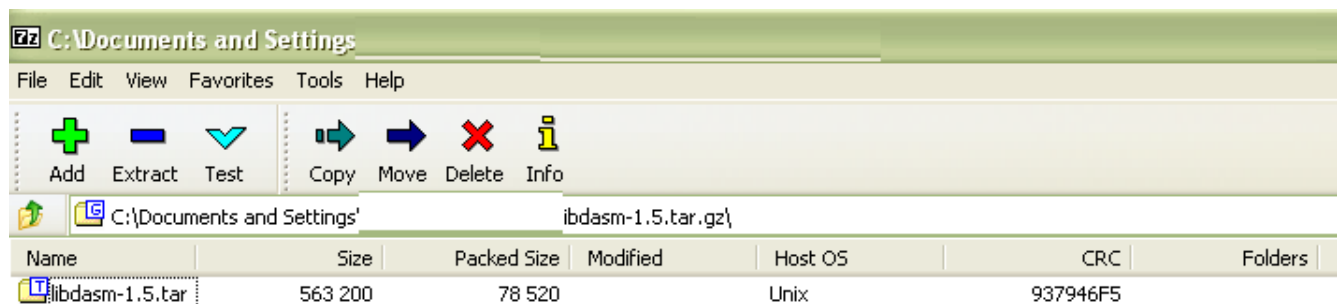


Figure 14: Opening libdasm tar ball with 7-zip

Double click on the libdasm*.tar file inside from within the 7-zip application until you see a folder icon with the name libdasm-1.5 (or other version number):

Installing Volatility Plugins (Windows)

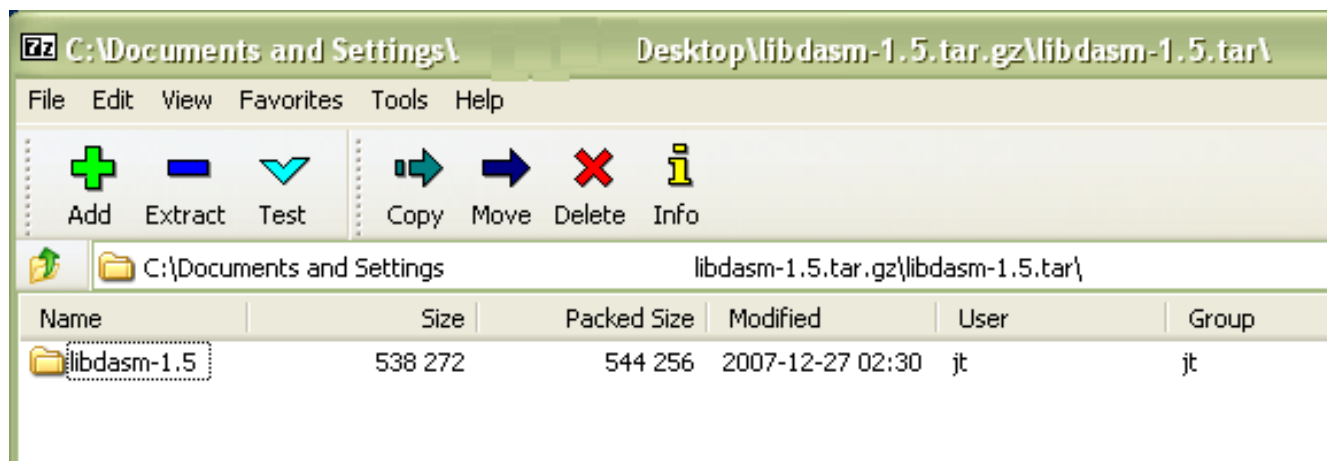


Figure 15: libdasm folder

Highlight the folder and then click on the extract button and say OK. The folder will extract with all source code inside to the path you choose, or by default the current directory:

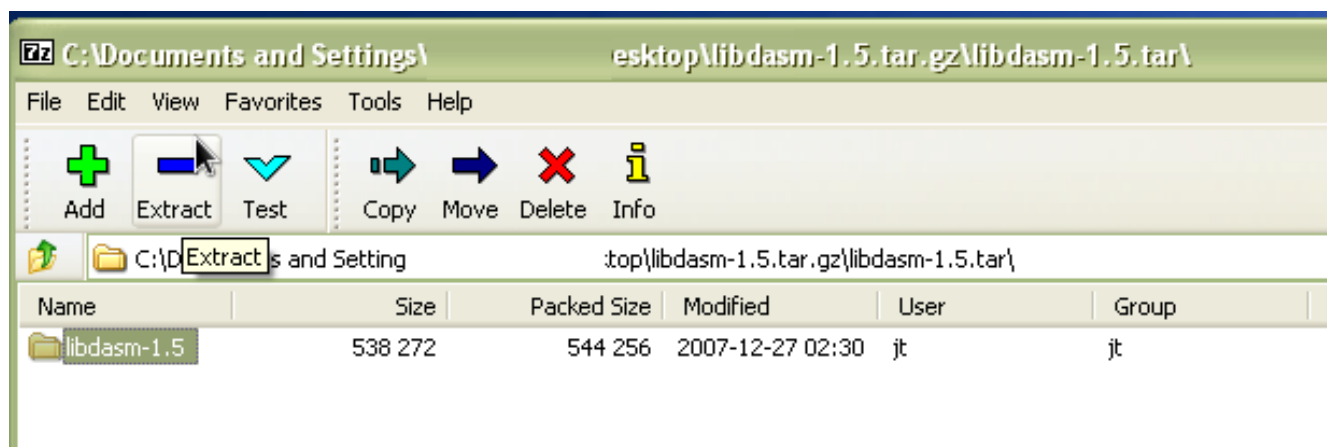


Figure 16: Extracting the libdasm source code

Now open a command prompt and change directories until you are in the newly extracted **libdasm** folder. Type the following commands:

```
make
cd pydasm
python setup.py build -c mingw32
python setup.py install
```

That's it! You've installed pydasm.

Installing Volatility Plugins (Windows)

Now you are ready to install the [pefile](#) library. Grab the zip file or tar ball of the source code and extract it as you did before. Go into that resulting folder and type the following:

```
python setup.py build
python setup.py install
```

Now you've installed pefile. Now you should see the malfind plugin listed under supported plugins for Volatility. All the other plugins that were depend on pefile should work as well if they are installed in the **memory_plugins** directory.

Installing the [volreg](#) plugin requires [pycrypto](#). Simply go the [gitweb interface](#) for this project and download the latest git snapshot by clicking on "snapshot". This will download a tar ball file of the source code. Simply extract it as you did before, then open the command prompt and change into that directory. Then type the following:

```
python setup.py build
python setup.py install
```

You've now installed the pycrypto library. Download the [volreg tarfile](#) and extract the contents into your Volatility folder by double clicking as before, selecting all three folders and changing the extraction path to your Volatility folder. All files should be placed into the correct location:

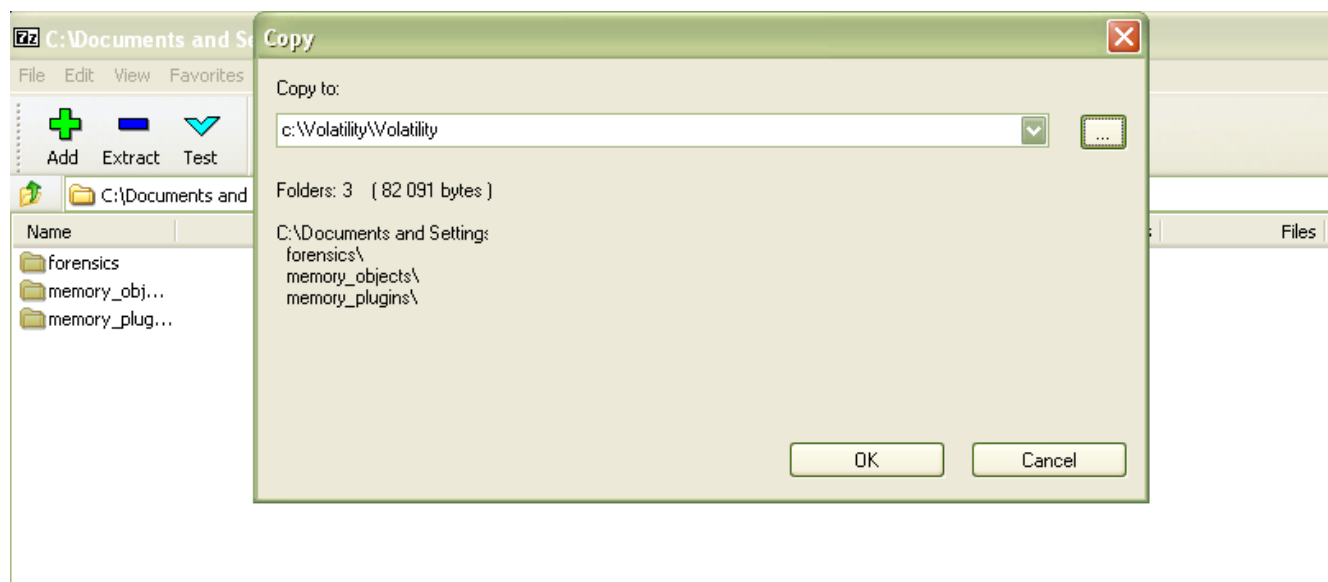


Figure 17: Extraction of volreg into Volatility directory.

Next time we will cover the [volrip plugin](#) after I figure out how to get Inline::Python working under windows...