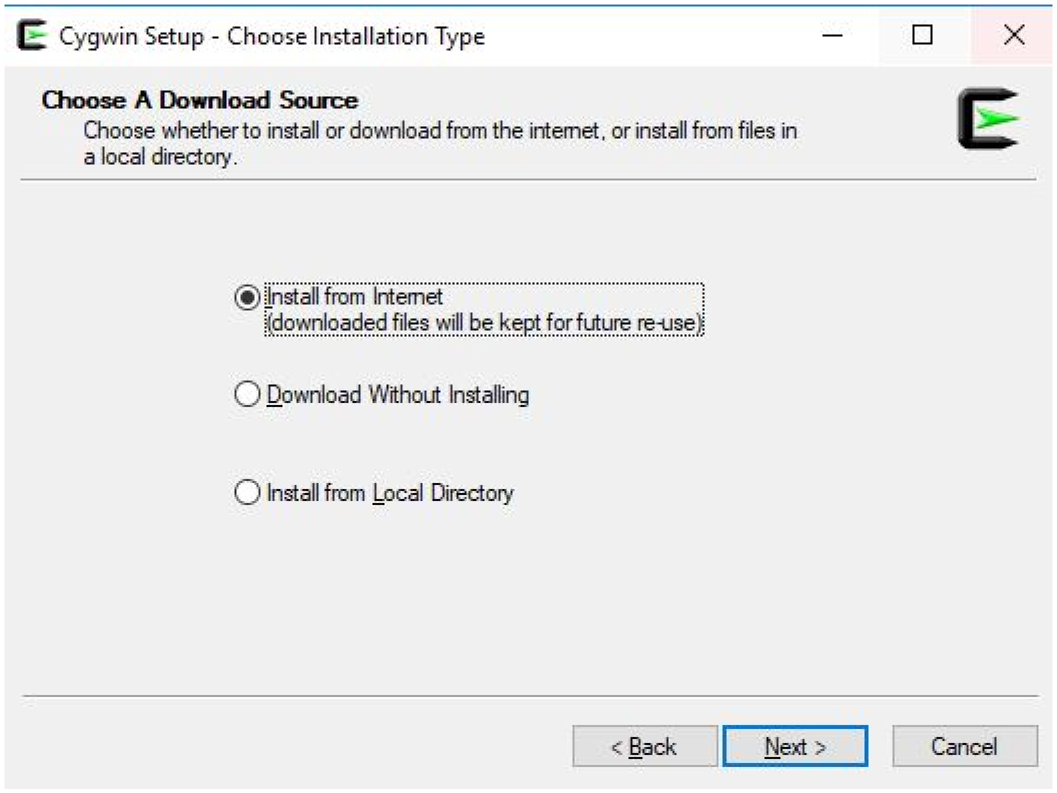
The document will bring the user through the proccess of installing both Cygwin and MAVProxy to be used alongside Mission Planner. The instructions below are can also be found in the following links, these links are for Windows machines:

- `http://ardupilot.org/dev/docs/building-setup-windows-cygwin.html`
- `http://ardupilot.org/dev/docs/sitl-native-on-windows.html`

Installing Cygwin

To start first download the executable for cygwin using this link for Windows 64bit `https://www.cygwin.com/setup-x86_64.exe` or `https://www.cygwin.com/setup-x86.exe` for Windows 32bit. Once it has downloaded run the installer, when it opens select next. Then make sure that `Install from internet` is selected as shown below in Figure 1, then click next.


Cygwin Setup - Choose Installation Type

Choose A Download Source

Choose whether to install or download from the internet, or install from files in a local directory.

● Install from Internet (downloaded files will be kept for future re-use)

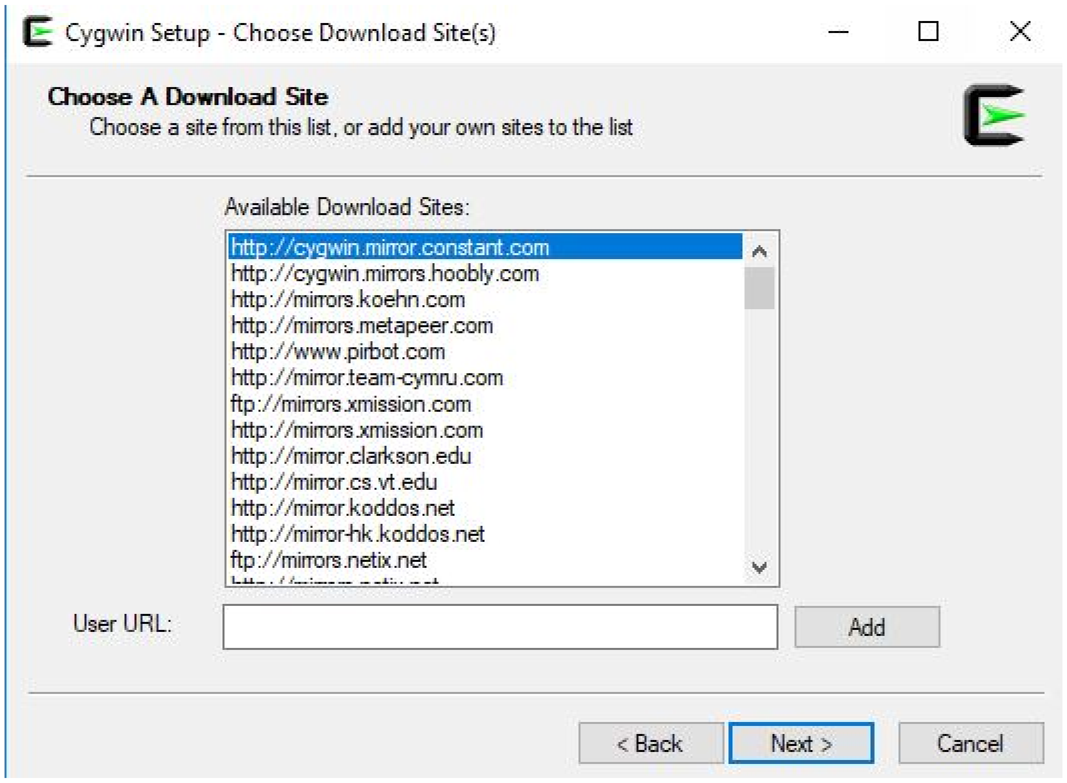
○ Download Without Installing

○ Install from Local Directory

< Back | Next > | Cancel

Figure 1: Install from internet selected

Next it will ask you to select a root directory, by default it may look like `C:\cygwin64` but you can chose for it to be anywhere. To prevent issues with errors it is recommended to keep the path short. Once the root directory has been selected click next. Now it should ask you for a location for the Local Package Directory, the directory will hold the binaries. By default is may look like `C:\User\username\Downloads`, I recommend moving it out of the Downloads directory to something like the Documents or Program Data directories ex. `C:\ProgramData`, `C:\User\username\Documents`. Once you have selected a directory click next and it should as you for your internet connection, select `Use System Proxy Settings` and click next. Next it will ask to `Choose A Download Site`. Select the first item as shown below in Figure 2 and click next.



Choose A Download Site

Choose a site from this list, or add your own sites to the list

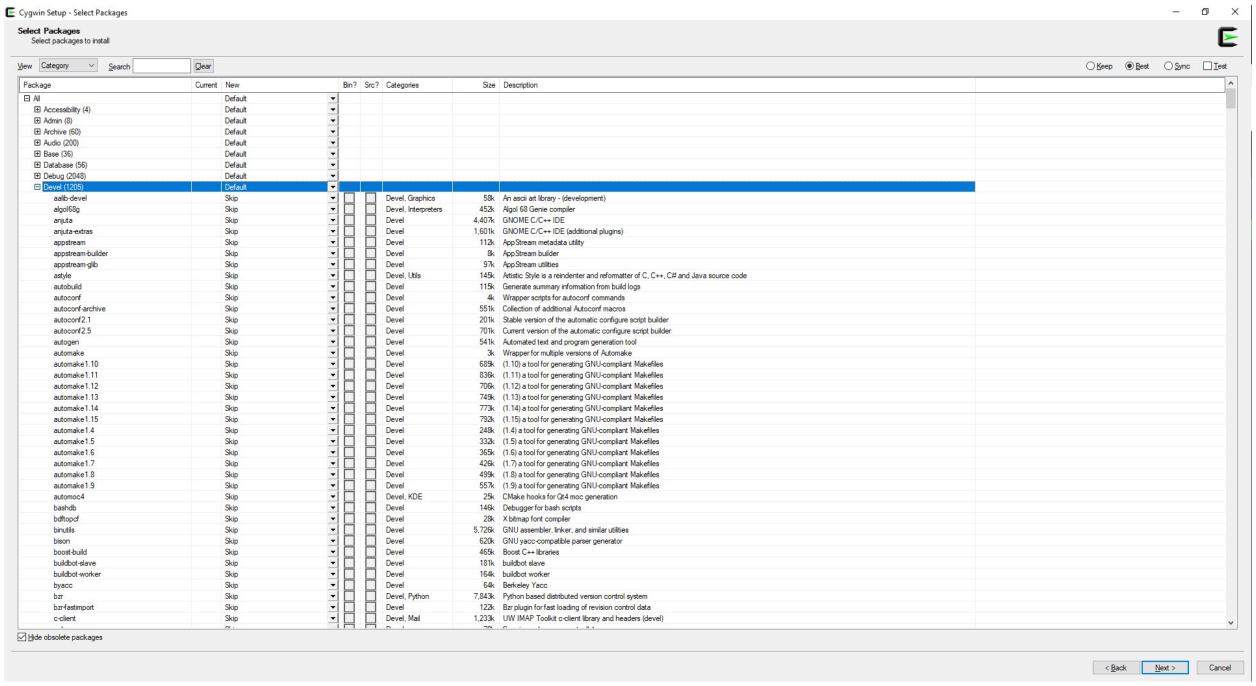
Available Download Sites:

http://cygwin.mirror.constant.com
http://cygwin.mirrors.hoobly.com
http://mirrors.koehn.com
http://mirrors.metapeer.com
http://www.pirbot.com
http://mirror.team-cymru.com
ftp://mirrors.xmission.com
http://mirrors.xmission.com
http://mirror.clarkson.edu
http://mirror.cs.vt.edu
http://mirror.koddos.net
http://mirror-hk.koddos.net
ftp://mirrors.netix.net

User URL:

Figure 2: Selecting a download site

Next you should get a screen looks like Figure 3. In the top left hand corner make sure that `View` category is `Category` and click the plus by the package directory to expand it. To find packages I recommend using the `Search` feature next to the `View` category. Table 1 will include all the packages required while Table 2 will include items recommended to add. All packages can be installed at a later time by rerunning the setup executable. To selected the package click `Skip` under the `New` column an select the most current version. Do not select the test version. Once you have selected all of the packages you need, click next and next again and it should start to install. The installation may take a few minutes.



Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Category Search: ( ) Clear   ( ) Keep [✓ Best] ( ) Sync ☐ Test

Package	Current	New	Bin?	Src?	Categories	Size	Description
⊟ All		Default					
⊞ Accessibility (4)		Default					
⊞ Admin (8)		Default					
⊞ Archive (60)		Default					
⊞ Audio (200)		Default					
⊞ Base (36)		Default					
⊞ Database (56)		Default					
⊞ Debug (2048)		Default					
⊟ Devel (1205)		Default					
aalib-devel		Skip	☐	☐	Devel, Graphics	58k	An ascii art library - (development)
algol68g		Skip	☐	☐	Devel, Interpreters	452k	Algol 68 Genie compiler
anjuta		Skip	☐	☐	Devel	4,407k	GNOME C/C++ IDE
anjuta-extras		Skip	☐	☐	Devel	1,601k	GNOME C/C++ IDE (additional plugins)
appstream		Skip	☐	☐	Devel	112k	AppStream metadata utility
appstream-builder		Skip	☐	☐	Devel	8k	AppStream builder
appstream-glib		Skip	☐	☐	Devel	97k	AppStream utilities
astyle		Skip	☐	☐	Devel, Utils	145k	Artistic Style is a reindenter and reformatter of C, C++, C# and Java source code
autobuild		Skip	☐	☐	Devel	115k	Generate summary information from build logs
autoconf		Skip	☐	☐	Devel	4k	Wrapper scripts for autoconf commands
autoconf-archive		Skip	☐	☐	Devel	551k	Collection of additional Autoconf macros
autoconf2.1		Skip	☐	☐	Devel	201k	Stable version of the automatic configure script builder
autoconf2.5		Skip	☐	☐	Devel	701k	Current version of the automatic configure script builder
autogen		Skip	☐	☐	Devel	541k	Automated text and program generation tool
automake		Skip	☐	☐	Devel	3k	Wrapper for multiple versions of Automake
automake1.10		Skip	☐	☐	Devel	689k	(1.10) a tool for generating GNU-compliant Makefiles
automake1.11		Skip	☐	☐	Devel	836k	(1.11) a tool for generating GNU-compliant Makefiles
automake1.12		Skip	☐	☐	Devel	706k	(1.12) a tool for generating GNU-compliant Makefiles
automake1.13		Skip	☐	☐	Devel	749k	(1.13) a tool for generating GNU-compliant Makefiles
automake1.14		Skip	☐	☐	Devel	773k	(1.14) a tool for generating GNU-compliant Makefiles
automake1.15		Skip	☐	☐	Devel	792k	(1.15) a tool for generating GNU-compliant Makefiles
automake1.4		Skip	☐	☐	Devel	248k	(1.4) a tool for generating GNU-compliant Makefiles
automake1.5		Skip	☐	☐	Devel	332k	(1.5) a tool for generating GNU-compliant Makefiles
automake1.6		Skip	☐	☐	Devel	365k	(1.6) a tool for generating GNU-compliant Makefiles
automake1.7		Skip	☐	☐	Devel	426k	(1.7) a tool for generating GNU-compliant Makefiles
automake1.8		Skip	☐	☐	Devel	499k	(1.8) a tool for generating GNU-compliant Makefiles
automake1.9		Skip	☐	☐	Devel	557k	(1.9) a tool for generating GNU-compliant Makefiles
automoc4		Skip	☐	☐	Devel, KDE	25k	CMake hooks for Qt4 moc generation
bashdb		Skip	☐	☐	Devel	146k	Debugger for bash scripts
bdftopcf		Skip	☐	☐	Devel	28k	X bitmap font compiler
binutils		Skip	☐	☐	Devel	5,726k	GNU assembler, linker, and similar utilities
bison		Skip	☐	☐	Devel	620k	GNU yacc-compatible parser generator
boost-build		Skip	☐	☐	Devel	465k	Boost C++ libraries
buildbot-slave		Skip	☐	☐	Devel	181k	buildbot slave
buildbot-worker		Skip	☐	☐	Devel	164k	buildbot worker
byacc		Skip	☐	☐	Devel	64k	Berkeley Yacc
bzr		Skip	☐	☐	Devel, Python	7,843k	Python based distributed version control system
bzr-fastimport		Skip	☐	☐	Devel	122k	Bzr plugin for fast loading of revision control data
c-client		Skip	☐	☐	Devel, Mail	1,233k	UW IMAP Toolkit c-client library and headers (devel)

☑ Hide obsolete packages

< Back Next > Cancel

Figure 3: Package selection

Name	Category / Name / Description
autoconf	Devel | autoconf: Wrapper scripts for autoconf commands
automake	Devel | automake: Wrapper scripts for automake and aclocal
ccache	Devel | ccache: A C compiler cache for improving recompilation
gcc-g++	Devel | gcc-g++ GNU Compiler Collection (C++)
git	Devel | git: Distributed version control system
libtool	Devel | libtool: Generic library support script
make	Devel | make: The GNU version of the 'make' utility
gawk	Interpreters | gawk: GNU awk, a pattern scanning and processing language
libexpat-devel	Libs | libexpat-devel: Expat XML parser library (development files)
libxml2-devel	Libs | libxml2-devel: Gnome XML library (development)
libxslt-devel	Libs | libxslt-devel: XML template library (development files)
libcrypt-devel	Libs | Encryption/Decryption utility and library
python2-devel	Python | python2-devel: Python2 language interpreter (python3 does not work yet)
python2-libxml2	Python | python2-libxml2: Gnome XML Library (Python bindings)
python27-pip	Python | python27-pip: Python package installation tool
procps-ng	System | procps-ng: System and process monitoring utilities (required for pkill)
gdb	Devel | gdb: The GNU Debugger
ddd	Devel | ddd: DDD, the data display debugger
zip	Archive | zip: Info-ZIP compression utility
nano	Devel | unzip: Info-ZIP decompression utility

Table 1: Required Packages

Name	Category / Name / Description
patch	Devel | patch: Applies diff files
cmake	Devel | cmake: Cross-platform makefile generation system
flex	Devel | flex: A fast lexical analizer generator
bison	Devel | bison: GNU yacc-compatible parser generator
unzip	Devel | unzip: Info-ZIP decompression utility
emacs	Devel | unzip: Info-ZIP decompression utility

Table 2: Recommended Packages

After the installation is complete click next finish. Once that is complete open up Cygwin. Once it is opened type in the following commands in the prompt. The installation of `pymavlink` may take some time. If you run into an error with one of the steps you may not be able to continue until it is resolved. If you run into an error make sure that you installed all the required items in the previous step, if you continue to have issues you may contact me.

```
pip2 install --upgrade pip
pip2 install argparse
pip2 install empy
pip2 install pyserial
```

```
pip2 install pymavlink
python -m ensurepip --user
python -m pip install --user future
python -m pip install --user lxml
python -m pip install --user uavcan
```

If you are able to complete the previous step without errors you will be ready to get the ardupilot git repository. To get the repository follow the commands below.

```
git clone git://github.com/ArduPilot/ardupilot.git
cd ardupilot
git submodule update --init --recursive
```

Once that is complete we need to add autotest to the pathway variables. To do this you will need to locate you `username.bashrc` file, this should be found in your home directory. At the end of the file add `export PATH=$PATH: /ardupilot/Tools/autotest`

```
cd ..
nano .bashrc
```

Once you have that added to the file save it and exit. At this point if you have not encounter any errors you are ready to install MAVProxy.

Installing MAVProxy

To install MAVProxy use the following link (for Windows machines) `http://firmware.ardupilot.org/Tools/MAVProxy/MAVProxySetup-latest.exe` for the latest version of MAVProxy. When the executable finishes downloading, click on it to run it and click yes. Then a License Agreement will appear, click I accept the agreement and then click next. Next it will ask you to select a destination location, again you may choose anywhere but it is recommended to keep the pathway short. Click next and click next again. Select the box to create a desktop shortcut and click next and install. Now MAVProxy will install and when it is done click finish. You do not need to open it right away. Once it is installed it also needs to be added to the pathway variables. So again add the following to the end of your .bashrc file. `export PATH=$PATH:location/of/install/MAVProxy` then save and exit the file. Once you have exited the file exit cygwin and reopen it.

```
nano .bashrc
```

Once it has been opened you can not simulate a aircraft. Enter the directory of the type of vehicle you want to simulate ex. `ArduCopter`, `ArduPlane` and run the following.

```
cd ardupilot/ArduPlane
sim_vehicle.py --map --console
```

At this cygwin will start running may proccesses, this may take quite a bit of time but by when it is complete it should automatically open up MAVProxy you completed all of the steps properly with no errors. For more on how to use MAVProxy please refer to the following link `http://ardupilot.org/dev/docs/sitl-native-on-windows.html`. If you have any issues installing or run into any errors you can not solve you may contact me at aes2@iastate.edu.