Daniel Parrott
2019 January 12

Building the Windows and DOS versions of FindOrb

Background

There is very little documentation on how to build the FindOrb software on Windows and DOS. This is largely due to the fact that most people only require the pre-built binaries, but unfortunately only the Windows binaries are available. The DOS binaries are useful for performing batch processing, and more importantly, being able to generate a virtual 'MPCORB' file from a list of observations. After spending an afternoon and some trial-and-error, I was finally able to build both the Windows and DOS executables from source.

Prerequisites

Before continuing, you will need the following items:

- Microsoft Visual Studio 2015 (Community Edition is free)
- The source from GitHub, or from another mirror of the source

From GitHub

If you want to work from the GitHub repository (recommended), then here are the links to retrieve the necessary dependencies (FindOrb has several):

1. https://github.com/Bill-Gray/lunar
2. https://github.com/Bill-Gray/jpl_eph
3. https://github.com/Bill-Gray/sat_code
4. https://github.com/Bill-Gray/miscell
5. https://github.com/Bill-Gray/find_orb
6. https://github.com/wmcbrine/PDCurses

For each of the above items, you will click on the "Clone or Download" button, and then choose "Download ZIP". Effectively, you should wind up with the following files:

1. lunar-master.zip
2. jpl_eph-master.zip
3. sat_code-master.zip
4. miscell-master.zip
5. find_orb-master.zip
6. PDCurses-master.zip

Place all of these zip files into a new directory for this project, like so:
..\find_orb-source\

1. ..\find_orb-source\lunar-master.zip
2. ..\find_orb-source\jpl_eph-master.zip
3. ..\find_orb-source\sat_code-master.zip
4. ..\find_orb-source\miscell-master.zip
5. ..\find_orb-source\find_orb-master.zip
6. ..\find_orb-source\PDCurses-master.zip

Proceed to extract all of the ZIP files.

Now, the first item to build will be “PDCurses”.

For this item, and every other item, you will be using the “Developer Command Prompt for VS2015”. This is like the regular command prompt, but it sets certain environment variables to make everything work. You should find it under your Visual Studio 2015 installation.

Once you have opened the “Developer” command prompt, proceed to navigate to the “PDCurses” directory and build the makefile:

> cd PDCurses-master
> cd wincon
> nmake -f Makefile.vc

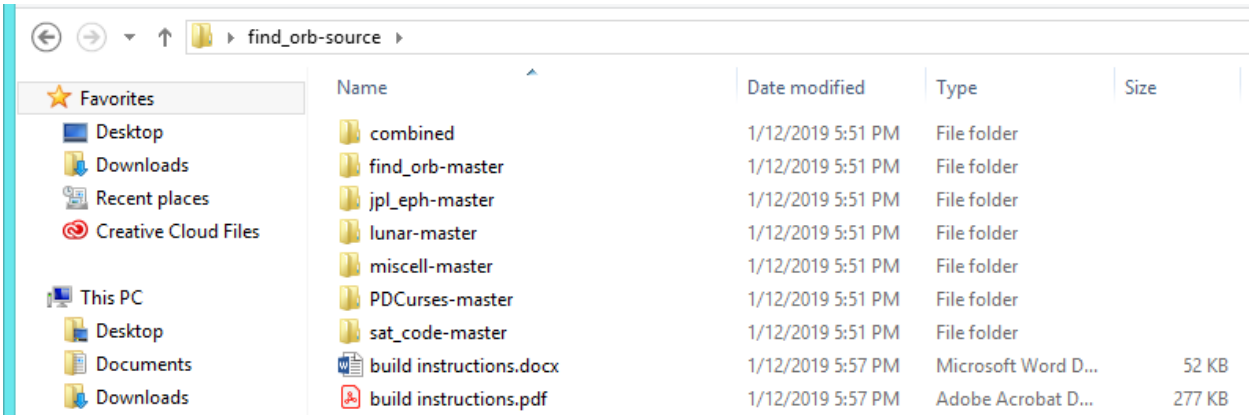
This will produce a library file, “pdcurses.lib”. Copy this file into a new directory as follows:

..\find_orb-source\combined\pdcurses.lib

(In other words, create a new directory, “combined”, and copy pdcurses.lib into it).

Next, you will also want to copy the “curses.h” file, located in “PDCurses-master”, into the “combined” directory as well:
..\find_orb-source\combined\curses.h

This sets the stage for everything else to be built, as we will eventually copy all of the other project files into this “combined” directory:



But first, we need to fix the “lunar” project, as it has some issues.

If you examine the “lunar.def” and “lunar64.def” files (under lunar-master) you will notice that the “lunar.def” file exports more symbols. This is a problem because we are building 64-bit and we want all of those symbols to be exported. See this page for someone else who ran into the problem as well:

https://github.com/Bill-Gray/lunar/issues/2

So, what I did was delete “lunar64.def”, and copy “lunar.def” to “lunar64.def”. Then I changed the top line:

from
“LIBRARY lunar”
to
“LIBRARY lunar64”

and saved the “lunar64.def” file with the above change. Both .def files are now identical except for the top line.

The other issue I ran into is that the “lunar-master” project has its own “miscell.cpp” file and so does the “find_orb-master” project. So we need to differentiate between the two by renaming one of them. I therefore decided to rename the “miscell.cpp” file in “lunar-master”

from
“miscell.cpp”
to
“miscell-lunar.cpp”

Doing this necessitates a change to the “lunar.mak” file as well. You will want to find and replace “miscell.obj” to “miscell-lunar.obj”.

With these three changes (lunar64.def, miscell-lunar.cpp, and miscell-lunar.obj) it is now time to copy the contents of the “lunar-master” directory into the “combined” directory.

Next, inside the Developer command prompt, navigate to the “combined” directory and proceed to build the lunar.mak file:

> nmake -f lunar.mak

You now have a “lunar64.lib” file.

The next item to build is “jpl_eph”. Proceed to copy the entire contents of its directory “jpl_eph-master” into the “combined” directory just as you did for the lunar project. If prompted about replacing files, such as “README” or “LICENSE” or “makefile” or “watcom.mak”, etc, those files do not matter, so you can replace them.

Now, again inside the “combined” directory, build the jpl_eph item, which uses a makefile called “vc.mak”

> nmake -f vc.mak

You now have a “jpleph.lib” file.

The next item to build is “sat_code”. For some reason, the makefile for this item, “msvc.mak”, needed several changes in order for it to work. I cannot imagine why it was shipped in such a broken state, but here is what you need to do to fix it.

First, examine the following section near the top of the “msvc.mak” file:

```
!ifdef BITS_32
COMMON_FLAGS=-nologo -W3 -EHsc -c -FD
RM=rm
!else
COMMON_FLAGS=-nologo -W3 -EHsc -c -FD -D_CRT_SECURE_NO_WARNINGS
RM=del
!endif
```

You will make two insertions, both of them immediately following the “COMMON_FLAGS” lines.

After the first “COMMON_FLAGS” line, make the following insertion:

```
LIBNAME=lunar
```

After the second “COMMON_FLAGS” line, make the following insertion:

```
LIBNAME=lunar64
```

The revised section should now look like this:

```
!ifdef BITS_32
COMMON_FLAGS=-nologo -W3 -EHsc -c -FD
LIBNAME=lunar
RM=rm
!else
COMMON_FLAGS=-nologo -W3 -EHsc -c -FD -D_CRT_SECURE_NO_WARNINGS
LIBNAME=lunar64
RM=del
!endif
```

One more modification is necessary. Find the following section:

```
sat_id.exe: sat_id.obj observe.obj sat_code.lib
  $(LINK)  sat_id.obj observe.obj sat_code.lib
```

And change to this:

```
sat_id.exe: sat_id.obj observe.obj sat_code.lib
  $(LINK)  sat_id.obj observe.obj sat_code.lib $(LIBNAME).lib
```

In other words, you are saying that this item (sat_id.exe) requires the lunar64.lib library (or lunar.lib if doing a 32-bit build).

Having fixed the “msvc.mak” file, you will now copy the entire contents of the “sat_code-master” directory into the “combined” directory, just as you did for the “lunar” and “jpl_eph” projects.

Proceed to build the sat_code item:

> nmake -f msvc.mak

You now have a “sat_code.lib” file.

Next, proceed to copy the entire contents of the “miscell-master” directory into the “combined” directory. There is no makefile for this item under Windows or DOS, so we will proceed to the last project, “find_orb” itself.

Before copying the contents of the “find_orb-master” directory into the “combined” directory, note that there is already an “elem2tle.cpp” file in both directories. To fix this, rename the version in the “find_orb-master” directory to “elem2tle-fo.cpp”. This will necessitate updating the two makefiles “win_find.mak” and “dos_find.mak” (which are also in the “find_orb-master” directory). In both files, search for “elem2tle.obj” and change it to “elem2tle-fo.obj”.

Also, both directories (“find_orb-master” and “combined”) have a “geo_rect.txt” file. I elected to copy the file from “find_orb-master” since that version is 24.6KB in size versus the 24.3KB in the “combined” directory. You will note that the “geo_rect.txt” file was originally copied from the “miscell-master” directory.

You can now proceed to copy over the entire contents of the “find_orb-master” directory into the “combined” directory.

Having copied the relevant files, make sure that the Developer command prompt is inside the “combined” directory, and proceed to build the “win_find.mak” file:

> nmake clean -f win_find.mak
> nmake -f win_find.mak

You now have “find_o64.exe” file.

We are almost finished. The next item is to build the DOS version:

> nmake clean -f dos_find.mak
> nmake -f dos_find.mak

You now have “fo.exe” file, which is the DOS version of FindOrb.

Congratulations, you have successfully built FindOrb from source on Windows.