

Simple Steps to Help You Install gVim on Windows 7

Vim is a text editor that is based on the Vi editor, which is common to UNIX-like systems. gVim is the GUI version of Vim. This article is a detailed account on how to install gVim on a Windows 7 machine along with Git, Pathogen and plugins.



I had used Vim on Linux earlier, but what I did was pretty basic. About two months ago, out of the blue, I installed it on Windows 7 on my work laptop and started trying it out. The initial days were filled with frustration and confusion but a few weeks later, I started getting comfortable. I am glad I stuck with it as a recent task at work involved verifying, editing, searching and replacing a lot of commands and scripts, and without gVim, it would have taken me at least twice as long.

Vim's power and flexibility is further enhanced by the plugins. All thanks to Git and Pathogen, installing plugins is now a breeze. However, there isn't any step-by-step guide on the Internet that shows you how to do it for gVim on Windows 7. I had to scour the Net, get information in bits and pieces and, often, there were just fleeting references on how to install it on Windows. It took me some time to figure things out and I initially wrote this article as a reference for myself. I kept adding steps to it and eventually thought it might also help others.

A note before we proceed. I am not at all a Vim expert. I just started using it about two months ago so the steps given below aren't exactly the 'one true way' of doing it. If you know a better way of doing things or feel that it can be improved, please let me know.

Just to ensure that the steps given here work fine, I completely uninstalled gVim, Git and Pathogen, and as I

reinstalled them, I took screenshots, captured the command line outputs and wrote the article.

The sequence of events

- 1) Download Vim and install it, move/delete a few folders (directories).
- 2) Next, install Git and Pathogen, which will make installing/uninstalling the Vim plugins easy. We'll also add a few settings in the `_vimrc` file.
- 3) Install a few colour scheme plugins.

Basically, when you install gVim, it will look like what's shown in Figure 1.

It's fairly plain and simple, and could use a better font and background; so let's tweak it and after we are done, it'll look like what's shown in Figure 2.

Note, Vim will work from the command prompt (Console version) but this article focuses on gVim.

gVim installation

You can download gVim from www.vim.org. In fact, right on the home page, you'll find a link to download the Windows version.

After downloading it, you can install it as usual in the `C:\Program Files (x86)` directory, but it's better to install it in your home directory. In my case, this is `C:\Users\Admin`. Create a folder called 'Vim' under your

```

VIM - Vi IMproved

        version 7.4
    by Bram Moolenaar et al.
Vim is open source and freely distributable

        Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version7<Enter> for version info

```

Figure 1: gVim first look

home directory, and point the installation to it. So in my case, it is `C:\Users\Admin\vim`.

Note: You can use the lowercase ‘v’ in ‘vim’. I just happen to like ‘Vim’.

Please remember to select the ‘Full’ option during installation as shown in Figure 3. After you are done with the installation, it’ll look like what’s shown in Figure 1. It’s now time to change a few things.

Shuffle the folders and move them around

When you now look under the Vim folder, you’ll find that it has:

- 1) A `vim74` folder, where the `gVim/vim` executable and other related files reside.
- 2) A file called `_vimrc`, which is the game changer Vim configuration file. In Linux, it’s named `.vimrc`.
- 3) Another folder called `vimfiles`. This is where we will make a couple of changes. It will have a lot of folders, all of which are empty.

Note: You can skip moving/deleting folders, and simply create two folders under ‘vimfiles’ - one will be ‘autoload’ and another ‘bundle’, but I prefer the following method. It works fine and reduces clutter.

First, delete all the folders in the ‘vimfiles’ folder, but don’t delete ‘vimfiles’ itself. Then, open the ‘vim74’ folder and you’ll find another folder under it called ‘autoload’. Cut and paste that folder under ‘vimfiles’ so that it now has only the ‘autoload’ folder. Next, create another folder in ‘vimfiles’ called ‘bundle’. Your ‘vimfiles’ folder will now have ‘autoload’ (which you moved from the ‘vim74’ folder) and ‘bundle’ folders.

If you just created new folders as mentioned above and did not delete/move the folders, you’ll have a lot more in there. Re-open gVim, to ensure that you are not getting any error messages. If you do, please reinstall gVim and follow the above steps.

Git and Pathogen: plugin power to gVim

To describe it very briefly, Git is a revision control system in which programmers can share their code. It’s much more

```

VIM - Vi IMproved

        version 7.4
    by Bram Moolenaar et al.
Vim is open source and freely distributable

        Become a registered Vim user!
type  :help register<Enter>  for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version7<Enter> for version info

```

Figure 2: gVim modified

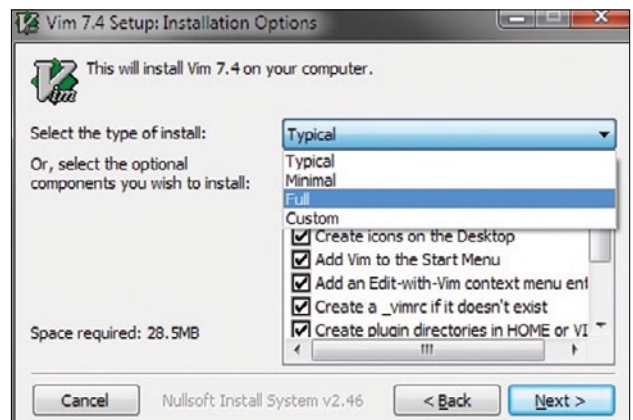


Figure 3: Select ‘Full’

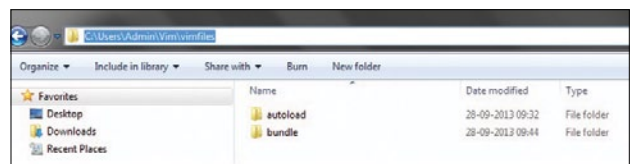


Figure 4: vimfiles

than that, though, and information about it can be obtained on the Internet.

Pathogen is what makes life easier for Vim users. It’s written by Tim Pope, the Vim and Vimscript expert. It makes installing or uninstalling plugins super easy. The Vim plugins and related files are spread across many Vim directories. Adding or removing them by hand could be both painful and dangerous. Thankfully, Pathogen makes it easier.

So let’s install Git and then Pathogen.

Head off to <http://git-scm.com/downloads> and download the Windows version. I’ve installed it in my home directory as that’s where I like it. That way, I just copy my home directory to a pen drive and use it anywhere I go.

There are two things you need to remember here. First, you must ensure that the ‘Git cheetah plug-in’ option is selected. It probably will be, by default, but it never hurts to make sure it is. The ‘True type font option for Windows Console’ selection is optional.

Second, during installation, ensure that the ‘Run Git from

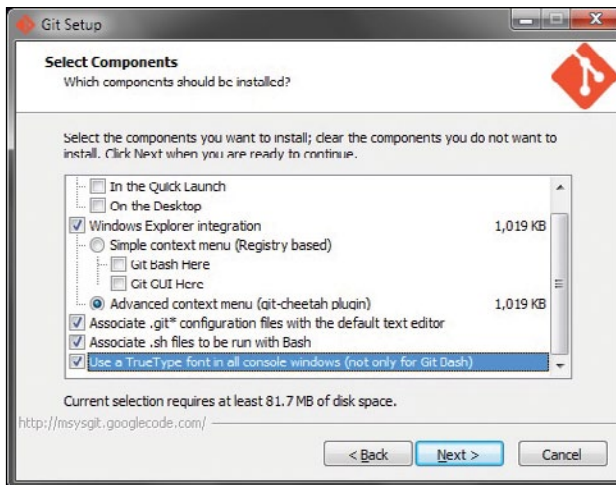


Figure 5: Select Git cheetah and true type

the Windows command prompt' option is selected.

Now, open the command prompt and type the 'git -version' command. You should see something like what is shown below:

```
C:\Users\Admin>git --version
git version 1.8.4.msysgit.0
```

This means Git is installed as expected. The good thing is that Curl is installed along with it, but we need to ensure that it can be run from the command prompt. So, let's head on to <https://github.com/gmarik/vundle/wiki/Vundle-for-Windows> and follow the instructions given under 'Curl on Windows'. A big 'thank you' to this guy (<https://github.com/gmarik>), as he has given clear instructions on how to install Git and Curl on Windows. He's also written 'Vundle', which can be used to install the plug-ins right within Vim, but more on that later (maybe some other article). Its guys like Tim Pope and 'gmarik' who are doing the amazing stuff that makes installing plugins easier. We all owe our thanks to them.

As shown in that article, copy paste the script and save it in a file called *curl.cmd* under the *cmd* directory, which is in the Git directory. In my case, the complete path to the *curl.cmd* file is as shown below:

```
C:\Users\Admin\Git\cmd\curl.cmd
```

Again, I installed it in my home directory; yours might show *C:\Program Files(x86)\Git* or something like that.

Once that's done, run the 'curl -version' command.

```
C:\Users\Admin>curl --version
curl 7.30.0 (i386-pc-win32) libcurl/7.30.0 OpenSSL/0.9.8y
zlib/1.2.7
Protocols: dict file ftp ftps gopher http https imap imaps
ldap ldaps pop3 pop3s rtsp smtp smtps telnet tftp
Features: AsynchDNS GSS-Negotiate IPv6 Largefile NTLM SPNEGO
```

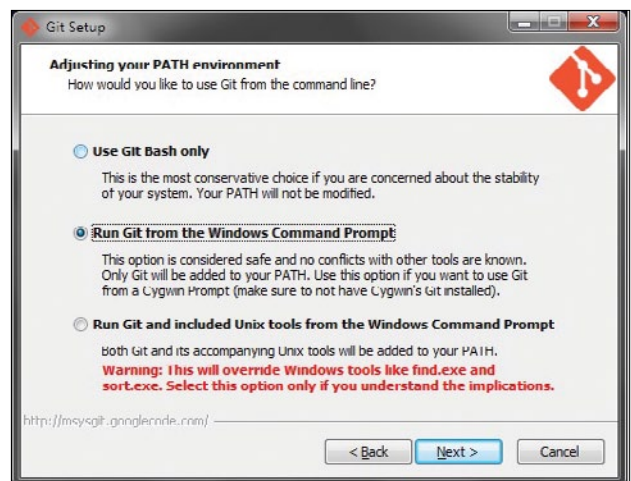


Figure 6: Run Git from Win prompt

SSL SSPI libz

Curl is now installed.

Let's install Pathogen. Time to *cd* to the autoload folder and run the Curl command as shown below:

```
C:\Users\Admin\Vim\vimfiles\autoload>curl -Sso pathogen.vim
https://raw.githubusercontent.com/tpope/vim-pathogen/master/autoload/pathogen.vim
```

A quick check to confirm:

```
C:\Users\Admin\Vim\vimfiles\autoload>dir | find /I "pathogen.vim"
28-09-2013 09:32          11,733 pathogen.vim
```

```
C:\Users\Admin\Vim\vimfiles\autoload>
```

As shown above, Pathogen is indeed installed.

Add settings to *_vimrc*

It's now time to add a few settings to *_vimrc*. Back it up first by either copying it to a different location, or to the same location with a different name. I have named it 'vimrc_backup'. This is the file we won't touch.

Now, open the *_vimrc* file. You'll see some content in there. Do not fiddle with it unless you know what you are doing. We will just append the following lines at the end of the *_vimrc*.

It's these lines that'll make Pathogen work and enable the plugins.

```
"Personal Settings.
"More to be added soon.
execute pathogen#infect()
filetype plugin indent on
syntax on
```

Plugin play!

It's now time to install a plugin using Pathogen! For starters, let's install a colour scheme called 'kolor'.

Note: Please ensure that you are in the 'bundle' directory and only then run the command as shown below. This is really important. If you are in the 'autoload' folder, either move back to 'bundle' or just open a new command prompt and move to 'bundle'. Then run the command:

```
git clone https://github.com/zeis/vim-kolor
```

```
C:\Users\Admin\vim\vimfiles\bundle>git clone https://github.com/zeis/vim-kolor
Cloning into 'vim-kolor'...
remote: Counting objects: 67, done.
remote: Compressing objects: 100% (34/34), done.
Uremote: Total 67 (delta 21), reused 66 (delta 21)
Unpacking objects: 100% (67/67), done.
Checking connectivity... done
```

```
C:\Users\Admin\vim\vimfiles\bundle>
```

If all goes well, you should see something similar to the snippet above.

How did I get that path? I searched Github for Vim colour schemes, clicked on one of the search results, and copy pasted the URL after the 'git clone' command.

If you prefer a light background colour scheme, try out 'eclipse' using the methods mentioned above. Another one is called 'Molokai'. Try that out too. Now, reopen gVim and type the following in 'command mode':

```
:colorscheme kolor
```

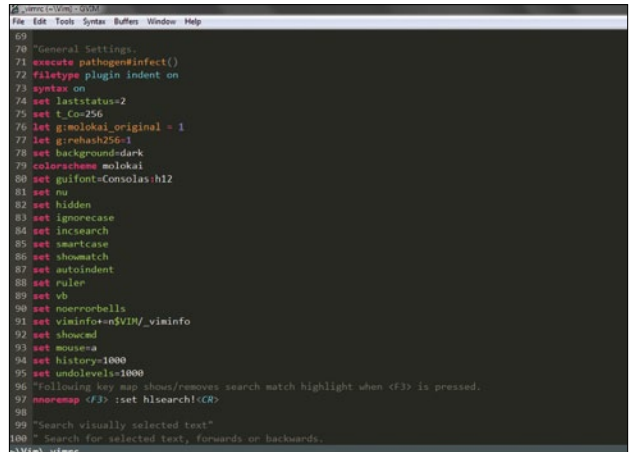
Similarly, try ':colorscheme eclipse' and/or ':colorscheme molokai'.

Tab completion works in command mode so you don't have to type in the whole thing.

gVim will forget the settings the moment you close it. Let's tweak our `_vimrc` further so that the settings are maintained, and while we are at it, add a little more functionality.

Add the following lines to the `_vimrc`, just below the lines we added earlier so that it looks like the snippet below. Let's set the colour scheme, font and other settings. This is an exact replica of my `_vimrc` and I've been using it for about a couple of days without any issues, so this should work. If you need more information on any of the settings, either type ':h setting' in the gVim command prompt or search on the Net.

```
"Personal Settings.
"More to be added soon.
execute pathogen#infect()
filetype plugin indent on
syntax on
```



```
vimrc
69
70 "General settings.
71 execute pathogen#infect()
72 filetype plugin indent on
73 syntax on
74 set laststatus=2
75 set t_Co=256
76 set gmlolaiOriginal = 1
77 let g:rehash256=1
78 set background=dark
79 colorscheme molokai
80 set guifont=Consolas:h12
81 set nu
82 set hidden
83 set ignorecase
84 set incsearch
85 set smartcase
86 set showmatch
87 set autoindent
88 set ruler
89 set vb
90 set noerrorbells
91 set viminfo+=n$VIM/_viminfo
92 set showcmd
93 set mouse=a
94 set history=1000
95 set undolevels=1000
96 "Following key map shows/removes search match highlight when (F3) is pressed.
97 nnoremap <F3> :set hsearch!<CR>
98
99 "Search visually selected text"
100 "Search for selected text, forwards or backwards.
```

Figure 7: vimrc in gVim

```
"Set the status line options. Make it show more information.
set laststatus=2
set statusline=%F%m%r%h%w\ [FORMAT=%{&ff}]\ [TYPE=%Y]\
[POS=%l,%v][%p%%]\ %{strftime("\%d/%m/%y\ -\ %H:%M")}
```


```
"Set Color Scheme and Font Options
```

```
colorscheme kolor
set guifont=Consolas:h12
```

```
"set line no, buffer, search, highlight, autoindent and more.
```

```
set nu
set hidden
set ignorecase
set incsearch
set smartcase
set showmatch
set autoindent
set ruler
set vb
set viminfo+=n$VIM/_viminfo
set noerrorbells
set showcmd
set mouse=a
set history=1000
set undolevels=1000
```

Save the `_vimrc`. Reopen gVim and type* ':e \$VIM/_vimrc'* in command mode.

If the options are added correctly and there are no conflicts, your gVim will now appear as shown in Figure 7, with a nice dark background and syntax highlights. **END** 

By: Pritesh Ugrankar

The author is an open source enthusiast who loves dabbling with Linux and other open source tools to find ways of automating boring repetitive tasks (but seldom succeeds at it). Drop him a line (or two) about any doubts, questions or opinions about this article at pmu.rwx@gmail.com