

Making Vim Run
|usr_90.txt| Installing Vim

=====

usr_90.txt For Vim version 8.0. Last change: 2008 Sep 10

VIM USER MANUAL - by Bram Moolenaar

Installing Vim

install

Before you can use Vim you have to install it. Depending on your system it's simple or easy. This chapter gives a few hints and also explains how upgrading to a new version is done.

|90.1| Unix
|90.2| MS-Windows
|90.3| Upgrading
|90.4| Common installation issues
|90.5| Uninstalling Vim

Previous chapter: |usr_45.txt| Select your language
Table of contents: |usr_toc.txt|

=====

90.1 Unix

First you have to decide if you are going to install Vim system-wide or for a single user. The installation is almost the same, but the directory where Vim is installed in differs.

For a system-wide installation the base directory "/usr/local" is often used. But this may be different for your system. Try finding out where other packages are installed.

When installing for a single user, you can use your home directory as the base. The files will be placed in subdirectories like "bin" and "shared/vim".

FROM A PACKAGE

You can get precompiled binaries for many different UNIX systems. There is a long list with links on this page:

<http://www.vim.org/binaries.html> ~

Volunteers maintain the binaries, so they are often out of date. It is a good idea to compile your own UNIX version from the source. Also, creating the editor from the source allows you to control which features are compiled. This does require a compiler though.

If you have a Linux distribution, the "vi" program is probably a minimal version of Vim. It doesn't do syntax highlighting, for example. Try finding another Vim package in your distribution, or search on the web site.

FROM SOURCES

To compile and install Vim, you will need the following:

- A C compiler (GCC preferred)
- The GZIP program (you can get it from www.gnu.org)
- The Vim source and runtime archives

To get the Vim archives, look in this file for a mirror near you, this should provide the fastest download:

```
ftp://ftp.vim.org/pub/vim/MIRRORS ~
```

Or use the home site `ftp.vim.org`, if you think it's fast enough. Go to the "unix" directory and you'll find a list of files there. The version number is embedded in the file name. You will want to get the most recent version.

You can get the files for Unix in two ways: One big archive that contains everything, or four smaller ones that each fit on a floppy disk. For version 6.1 the single big one is called:

```
vim-6.1.tar.bz2 ~
```

You need the `bzip2` program to uncompress it. If you don't have it, get the four smaller files, which can be uncompressed with `gzip`. For Vim 6.1 they are called:

```
vim-6.1-src1.tar.gz ~  
vim-6.1-src2.tar.gz ~  
vim-6.1-rt1.tar.gz ~  
vim-6.1-rt2.tar.gz ~
```

COMPILING

First create a top directory to work in, for example: >

```
mkdir ~/vim  
cd ~/vim
```

Then unpack the archives there. If you have the one big archive, you unpack it like this: >

```
bzip2 -d -c path/vim-6.1.tar.bz2 | tar xf -
```

Change "path" to where you have downloaded the file. >

```
gzip -d -c path/vim-6.1-src1.tar.gz | tar xf -  
gzip -d -c path/vim-6.1-src2.tar.gz | tar xf -  
gzip -d -c path/vim-6.1-rt1.tar.gz | tar xf -  
gzip -d -c path/vim-6.1-rt2.tar.gz | tar xf -
```

If you are satisfied with getting the default features, and your environment is setup properly, you should be able to compile Vim with just this: >

```
cd vim61/src  
make
```

The `make` program will run `configure` and compile everything. Further on we will explain how to compile with different features.

If there are errors while compiling, carefully look at the error messages. There should be a hint about what went wrong. Hopefully you will be able to correct it. You might have to disable some features to make Vim compile. Look in the Makefile for specific hints for your system.

TESTING

Now you can check if compiling worked OK: >

```
make test
```

This will run a sequence of test scripts to verify that Vim works as expected. Vim will be started many times and all kinds of text and messages flash by. If it is alright you will finally see:

```
test results: ~
ALL DONE ~
```

If you get "TEST FAILURE" some test failed. If there are one or two messages about failed tests, Vim might still work, but not perfectly. If you see a lot of error messages or Vim doesn't finish until the end, there must be something wrong. Either try to find out yourself, or find someone who can solve it. You could look in the `|maillist-archive|` for a solution. If everything else fails, you could ask in the vim `|maillist|` if someone can help you.

INSTALLING

`*install-home*`

If you want to install in your home directory, edit the Makefile and search for a line:

```
#prefix = $(HOME) ~
```

Remove the # at the start of the line.

When installing for the whole system, Vim has most likely already selected a good installation directory for you. You can also specify one, see below. You need to become root for the following.

To install Vim do: >

```
make install
```

That should move all the relevant files to the right place. Now you can try running vim to verify that it works. Use two simple tests to check if Vim can find its runtime files: >

```
:help
:syntax enable
```

If this doesn't work, use this command to check where Vim is looking for the runtime files: >

```
:echo $VIMRUNTIME
```

You can also start Vim with the "-V" argument to see what happens during startup: >

```
vim -V
```

Don't forget that the user manual assumes you Vim in a certain way. After installing Vim, follow the instructions at `|not-compatible|` to make Vim work as assumed in this manual.

SELECTING FEATURES

Vim has many ways to select features. One of the simple ways is to edit the Makefile. There are many directions and examples. Often you can enable or disable a feature by uncommenting a line.

An alternative is to run "configure" separately. This allows you to specify configuration options manually. The disadvantage is that you have to figure out what exactly to type.

Some of the most interesting configure arguments follow. These can also be enabled from the Makefile.

<code>--prefix={directory}</code>	Top directory where to install Vim.
<code>--with-features=tiny</code>	Compile with many features disabled.
<code>--with-features=small</code>	Compile with some features disabled.
<code>--with-features=big</code>	Compile with more features enabled.
<code>--with-features=huge</code>	Compile with most features enabled. See +feature-list for which feature is enabled in which case.
<code>--enable-perlinterp</code>	Enable the Perl interface. There are similar arguments for ruby, python and tcl.
<code>--disable-gui</code>	Do not compile the GUI interface.
<code>--without-x</code>	Do not compile X-windows features. When both of these are used, Vim will not connect to the X server, which makes startup faster.

To see the whole list use: >

```
./configure --help
```

You can find a bit of explanation for each feature, and links for more information here: |+feature-list|.

For the adventurous, edit the file "feature.h". You can also change the source code yourself!

```
=====
*90.2* MS-Windows
```

There are two ways to install the Vim program for Microsoft Windows. You can uncompress several archives, or use a self-installing big archive. Most users with fairly recent computers will prefer the second method. For the first one, you will need:

- An archive with binaries for Vim.
- The Vim runtime archive.
- A program to unpack the zip files.

To get the Vim archives, look in this file for a mirror near you, this should provide the fastest download:

```
ftp://ftp.vim.org/pub/vim/MIRRORS ~
```

Or use the home site ftp.vim.org, if you think it's fast enough. Go to the "pc" directory and you'll find a list of files there. The version number is embedded in the file name. You will want to get the most recent version. We will use "61" here, which is version 6.1.

```
gvim61.exe          The self-installing archive.
```

This is all you need for the second method. Just launch the executable, and follow the prompts.

For the first method you must chose one of the binary archives. These are available:

```
gvim61.zip          The normal MS-Windows GUI version.
```

gvim6l1ole.zip	The MS-Windows GUI version with OLE support. Uses more memory, supports interfacing with other OLE applications.
vim6lw32.zip	32 bit MS-Windows console version. For use in a Win NT/2000/XP console. Does not work well on Win 95/98.
vim6ld32.zip	32 bit MS-DOS version. For use in the Win 95/98 console window.
vim6ld16.zip	16 bit MS-DOS version. Only for old systems. Does not support long filenames.

You only need one of them. Although you could install both a GUI and a console version. You always need to get the archive with runtime files.

vim6l1rt.zip The runtime files.

Use your un-zip program to unpack the files. For example, using the "unzip" program: >

```
cd c:\
unzip path\gvim6l.zip
unzip path\vim6l1rt.zip
```

This will unpack the files in the directory "c:\vim\vim6l". If you already have a "vim" directory somewhere, you will want to move to the directory just above it.

Now change to the "vim\vim6l" directory and run the install program: >

```
install
```

Carefully look through the messages and select the options you want to use. If you finally select "do it" the install program will carry out the actions you selected.

The install program doesn't move the runtime files. They remain where you unpacked them.

In case you are not satisfied with the features included in the supplied binaries, you could try compiling Vim yourself. Get the source archive from the same location as where the binaries are. You need a compiler for which a makefile exists. Microsoft Visual C works, but is expensive. The Free Borland command-line compiler 5.5 can be used, as well as the free MingW and Cygwin compilers. Check the file src/INSTALLpc.txt for hints.

90.3 Upgrading

If you are running one version of Vim and want to install another, here is what to do.

UNIX

When you type "make install" the runtime files will be copied to a directory which is specific for this version. Thus they will not overwrite a previous version. This makes it possible to use two or more versions next to each other.

The executable "vim" will overwrite an older version. If you don't care about keeping the old version, running "make install" will work fine. You can delete the old runtime files manually. Just delete the directory with the version number in it and all files below it. Example: >

```
rm -rf /usr/local/share/vim/vim58
```

There are normally no changed files below this directory. If you did change the "filetype.vim" file, for example, you better merge the changes into the new version before deleting it.

If you are careful and want to try out the new version for a while before switching to it, install the new version under another name. You need to specify a configure argument. For example: >

```
./configure --with-vim-name=vim6
```

Before running "make install", you could use "make -n install" to check that no valuable existing files are overwritten.

When you finally decide to switch to the new version, all you need to do is to rename the binary to "vim". For example: >

```
mv /usr/local/bin/vim6 /usr/local/bin/vim
```

MS-WINDOWS

Upgrading is mostly equal to installing a new version. Just unpack the files in the same place as the previous version. A new directory will be created, e.g., "vim61", for the files of the new version. Your runtime files, vimrc file, viminfo, etc. will be left alone.

If you want to run the new version next to the old one, you will have to do some handwork. Don't run the install program, it will overwrite a few files of the old version. Execute the new binaries by specifying the full path. The program should be able to automatically find the runtime files for the right version. However, this won't work if you set the \$VIMRUNTIME variable somewhere.

If you are satisfied with the upgrade, you can delete the files of the previous version. See [90.5].

=====

90.4 Common installation issues

This section describes some of the common problems that occur when installing Vim and suggests some solutions. It also contains answers to many installation questions.

Q: I Do Not Have Root Privileges. How Do I Install Vim? (Unix)

Use the following configuration command to install Vim in a directory called \$HOME/vim: >

```
./configure --prefix=$HOME
```

This gives you a personal copy of Vim. You need to put \$HOME/bin in your path to execute the editor. Also see [install-home].

Q: The Colors Are Not Right on My Screen. (Unix)

Check your terminal settings by using the following command in a shell: >

```
echo $TERM
```

If the terminal type listed is not correct, fix it. For more hints, see [06.2]. Another solution is to always use the GUI version of Vim, called gvim. This avoids the need for a correct terminal setup.

Q: My Backspace And Delete Keys Don't Work Right

The definition of what key sends what code is very unclear for backspace <BS> and Delete keys. First of all, check your \$TERM setting. If there is nothing wrong with it, try this: >

```
:set t_kb=^V<BS>
:set t_kD=^V<Del>
```

In the first line you need to press CTRL-V and then hit the backspace key. In the second line you need to press CTRL-V and then hit the Delete key. You can put these lines in your vimrc file, see |05.1|. A disadvantage is that it won't work when you use another terminal some day. Look here for alternate solutions: |:fixdel|.

Q: I Am Using RedHat Linux. Can I Use the Vim That Comes with the System?

By default RedHat installs a minimal version of Vim. Check your RPM packages for something named "Vim-enhanced-version.rpm" and install that.

Q: How Do I Turn Syntax Coloring On? How do I make plugins work?

Use the example vimrc script. You can find an explanation on how to use it here: |not-compatible|.

See chapter 6 for information about syntax highlighting: |usr_06.txt|.

Q: What Is a Good vimrc File to Use?

See the www.vim.org Web site for several good examples.

Q: Where Do I Find a Good Vim Plugin?

See the Vim-online site: <http://vim.sf.net>. Many users have uploaded useful Vim scripts and plugins there.

Q: Where Do I Find More Tips?

See the Vim-online site: <http://vim.sf.net>. There is an archive with hints from Vim users. You might also want to search in the |maillist-archive|.

```
=====
*90.5*  Uninstalling Vim
```

In the unlikely event you want to uninstall Vim completely, this is how you do it.

UNIX

When you installed Vim as a package, check your package manager to find out how to remove the package again.

If you installed Vim from sources you can use this command: >

```
make uninstall
```

However, if you have deleted the original files or you used an archive that someone supplied, you can't do this. Do delete the files manually, here is an example for when "/usr/local" was used as the root: >

```
rm -rf /usr/local/share/vim/vim61
rm /usr/local/bin/evview
rm /usr/local/bin/evim
rm /usr/local/bin/ex
rm /usr/local/bin/gview
rm /usr/local/bin/gvim
rm /usr/local/bin/gvimdiff
rm /usr/local/bin/rgview
rm /usr/local/bin/rgvim
rm /usr/local/bin/rview
rm /usr/local/bin/rvim
rm /usr/local/bin/rvim
rm /usr/local/bin/view
rm /usr/local/bin/vim
rm /usr/local/bin/vimdiff
rm /usr/local/bin/vimtutor
rm /usr/local/bin/xxd
rm /usr/local/man/man1/evview.1
rm /usr/local/man/man1/evim.1
rm /usr/local/man/man1/ex.1
rm /usr/local/man/man1/gview.1
rm /usr/local/man/man1/gvim.1
rm /usr/local/man/man1/gvimdiff.1
rm /usr/local/man/man1/rgview.1
rm /usr/local/man/man1/rgvim.1
rm /usr/local/man/man1/rview.1
rm /usr/local/man/man1/rvim.1
rm /usr/local/man/man1/view.1
rm /usr/local/man/man1/vim.1
rm /usr/local/man/man1/vimdiff.1
rm /usr/local/man/man1/vimtutor.1
rm /usr/local/man/man1/xxd.1
```

MS-WINDOWS

If you installed Vim with the self-installing archive you can run the "uninstall-gui" program located in the same directory as the other Vim programs, e.g. "c:\vim\vim61". You can also launch it from the Start menu if installed the Vim entries there. This will remove most of the files, menu entries and desktop shortcuts. Some files may remain however, as they need a Windows restart before being deleted.

You will be given the option to remove the whole "vim" directory. It probably contains your vimrc file and other runtime files that you created, so be careful.

Else, if you installed Vim with the zip archives, the preferred way is to use the "uninstal" program (note the missing l at the end). You can find it in the same directory as the "install" program, e.g., "c:\vim\vim61". This should also work from the usual "install/remove software" page.

However, this only removes the registry entries for Vim. You have to delete the files yourself. Simply select the directory "vim\vim61" and delete it recursively. There should be no files there that you changed, but you might want to check that first.

The "vim" directory probably contains your vimrc file and other runtime files that you created. You might want to keep that.

=====

Table of contents: |usr_toc.txt|

Copyright: see |manual-copyright| vim:tw=78:ts=8:ft=help:norl: