U.S.R. O. three dot T.X.T.
for vim version eight point two
last change
twenty-twenty september O. three
vim user manual
by bram moolenaar
moving around
before you can insert or delete text the cursor has to be moved to the right place
vim has a large number of commands to position the cursor
this chapter shows you how to use the most important ones
you can find a list of these commands below
O. three point one
word movement
O. three point two
moving to the start or end of a line
O. three point three
moving to a character
O. three point four
matching a parenthesis
O. three point five
moving to a specific line
O. three point six
telling where you are
O. three point seven
scrolling around
O. three point eight
simple searches
O. three point nine
simple search patterns
O. three point ten
using marks
next chapter
U.S.R. underscore O. four T.X.T.
making small changes
previous chapters
U.S.R. underscore O. two dot T.X.T.
the first steps in vim
table of contents
U.S.R. underscore T.O.C. dot T.X.T.
O. three point one
word movement
to move the cursor forward one word
use the W. command
like most vim commands
you can use a numeric prefix to move past multiple words
for example
three W. moves three words
this figure shows how it works
starting at the position marked with X.
notice that W. moves to the start of the next word if it already is at the start of a word
the B. command moves backward to the start of the previous word
there is also the e command that moves to the next end of a word and G.E.
which moves to the previous end of a word
if you are at the last word of a line
the W. command will take to the first word in the next line

thus you can use this to move through a paragraph
much faster than using L.
B. does the same in the other direction
a word ends at a non word character
such as a period dash or closing parenthesis
to change what vim considers to be a word
see the is keyword option
if you try this out in the help directly
is keyword needs to be reset for the examples to work
colon set is keyword ampersand
it is also possible to move by white space separated words
this is not a word in the normal sense
that's why the uppercase is used
the commands for moving by words are also uppercase
as this figure shows
with this mix of lowercase and uppercase commands
you can quickly move forward and backward through a paragraph
O. three point two
moving to the start or end of a line
the dollar sign command moves the cursor to the end of a line
if your keyborad has an end key
it will do the same thing
the caret command moves to the first non blank character of the line
the zero command
zero
moves to the very first character of the line
and the home key does the same thing
in a picture
period indicates a space
the five periods indicates blanks here
the dollar sign command takes a count
like most movement commands
but moving to the end of the line several times doesn't make sense
therefore it causes the editor to move to the end of another line
for example
one dollar sign moves you to the end of the first line
the one you're on
two dollar sign to the end of the next line
and so on
the zero command doesn't take a count argument
because the 0 would be part of the count
unexpectedly
using a count with caret doesn't have any effect
O. three point three
moving to a character
one of the most useful movement commands is the single character search command
the command F.X. searches forward in the line for the single character X.
hint
F. stands for find
for example
you are at the beginning of the following line
suppose you want to go to the H. of human
just execute F.H. and the cursor
will be positioned over the H.

this also shows that the command F.Y. moves to the end of the word really
you can specify a count
therefore
you can go to the L. of foul with three. F.L.
the F. command searches to the left
the T.X. command works like the F.X. command
except it stops one character before the searched character
hint
T. stands for to
the backward version of this command is T.X.
to err is human
to really foul up you need a computer
these four commands can be repeated with semicolon
the comma repeapts in the other direction
the cursor is never moved to another line
not even when the sentence continues
sometimes you will start a search
only to realize that you have typed the wrogn command
you type F. to search backward
for example
only to realize that you really meant F.
to abort a search press escape
so F. escape is an aborted forward search and doesn't do anything
note
escape cancels most operations
not just searches
O. three point four
matching a parenthesis
when writing a program you often end up with nested parentheses constructs
then the percent sign command is very handy
it moves to the matching paren
if the cursor is on a opening parenthesis
it will move to the matching closing parenthesis
if it's on a closing parenthesis
it will move to the matching opening parenthesis
this also wrods for brackets and curly braces paris
this can be defined with the match pairs option
when the cursor is not on a useful character
percent sign will search forward to find one
thus if the cursor is at the start of the line of the previous example
percent sign will search forward and find the first opening parenthesis
then it moves to its match
O. three point five
moving to a specific line
if you are a C. or C. plus plus programmer
you are familiar with error messages such as the following
P.R.O.G. dot C. colon thirty-three colon J.
undeclared
first use in this function
this tells you that you might want to fix something on line thirty-three
so how do you find line thirty three?
one way is to do nine thousand nine hundred ninety-nine
to go to the top of the file and thirty-two J.
to go down thirty-two lines
it is not a good way

but it works
a much better way of doing things is to use the G. command
with a count
this command positions you at the given line number
for example
thirty-three uppercase G.
puts you on line thirty three
for a better way of going through a compiler's error list
see U.S.R. underscore thirty dot T.X.T.
for information on the colon make command
with no argument
uppercase G. positions you at the end of the file
a quick way to go th the start of a file use G.G.
one G. will do the same
but is a tiny bit more typing
another way to move to a line is using the percent sign command with a count
for example
fifty percent moves you halfway the file
ninety percent goes to near the end
the previous assumes that you want to move to a line in the file
no matter if it's currently visible or not
what if you want to move to one of the lines you can see?
this figures shows the three commands you can use
hints
H. stands for home
M. for middle
L. for last
alternatively
H. for high
M. for middle
and L. for low
O. three point six
telling where you are
to see where you are in a file
there are three ways
one
use the control G. command
you get a message like this
assuming the ruller option is off
U.S.R. underscore O. three dot T.X.T. line two hundred and thirty-three of six hundred and fifty
thirty-five percent column forty-five to fifty-two
this shows the name of the file you are editing
the line number where the cursor is
the total numbers of lines
the percentage of the way through the file and the column of the cursor
sometimes you will see a split column number
for example
column two to nine
this indicates that the cursor positione on the secone character
but because character one is a tab
occupying eight spaces worth of columns
the screen column is nine
two
set the number option
this will display a line number in front of every line

colon set number
to switch this off again
colon set no number
since number is a boolean option
prepending no to its name has the effect of switching it off
a boolean option has only these two values
it is either on or off
vim has many options
besides the boolean ones there are options with a numerical value and string options
you will see examples of this where they are used
three
set the ruler option
this will display the cursor position in the lower right corner of the vim windows
colon set ruler
using the ruler option has the advantage that it doesn't take much room
thus there is more space for your text
O. three point seven
scrolling around
the control U. command scrolls down half a screen of text
think of looking through a viewing window at the text and moving this window up by half the height of the window
thus the window moves up over the text
which is backward in the file
don't worry if you have a little trouble remembering which end is up
most users have the same problem
the control D. command moves the viewing window down half a screen in the file
thus scrolls the text up half a screen
to scroll one line at a time use control E.
scroll up
and control Y.
scroll down
think of control E.
to give you one line extra
if you use microsoft windows compatible key mappings
control Y. will redo a change instead of scroll
to scroll forward by a whole screen
except for two lines
use control F.
to scroll backwards
use control B.
these should be easy to remember
F. forwards and B. for backwards
a common issue is that after moving down many lines with J.
your cursor is at the bottom of the screen
you would like to see the context of the line with the cursor
that's done with the double Z. command
the Z.T. command puts the cursor line at the top
Z.B. at the bottom
there are a few more scrolling commands
see uppercase Q. underscore S.C.
to always keep a few lines of context around the cursor
use the scroll off option
O. three point eight
simple searches
to search for a string

use the slash string command
to find the word include
for example
use the command
slash include
you will notice that when you type the slash
the cursor jumps to the last line of the vim window
like with colon commands
that is where you type the word
you can press the backspace key
backarrow or backspace
to make corrections
use the left and right cursor keys when necessary
pressing enter executes the command
note
the character period asterisk opening bracket closing bracket caret percent sign slash backslash
question mark tilde dollar sign
have special meanings
if you want to use them in a search you must put a backslash in front of them
see below
to find the next occurrence of the same string use the N. command
use this to find the first hash sign include after the cursor
slash hash sign include
and then type N. several times
you will move to each hash sign include in the text
you can also use a count if you know which match you want
thus three N. finds the third match
you can also use a count with slash
four slash the
goes to the fourth match of the
the question mark command works like slash
but searches backwards
question mark word
the N command repeats the last search the opposite direction
thus using N. after a slash command searches backwards
using N. after question mark searches forwards
ignoring case
normally you have to type exactly what you want to find
if you don't care about upper or lowercase in a word
set the ignore case option
colon set ignore case
if you now search for word
it will also match word with the first character uppercase and word in all character uppercase
to match case again
colon set no ignore case
history
suppose you do three searches
slash one
slash two
slash three
now let's start searching by typing a simple slash without pressing enter
if you press up
the cursor key
vim puts slash three on the command line
pressing enter at this point searches for three

if you do not press enter
but press up instead
vim changes the prompt to slash two
another press of up moves you to slash one
you can also use the down cursor key to move through the history of search commands in the other direction
if you know what a previously used pattern starts with
and you want to use it again
type that character before pressing up
with the previous example
you can type slash O. up
and vim will put slash one on the command line
the command starting with colon also have a history
that allows you to recall a previous command and execute it again
these two histories are separate
searching for a word in the text
suppose you see the word
the long function name in camel case with the first character uppercase in the text and you want to find the next occurrence of it
you could type slash the long function name in camel case with the first character uppercase
but that's a lot of typing
and when you make a mistake vim won't find it
there is an easier way
position the cursor on the word and use the asterisk command
vim will grab the word under the cursor and use it as the search string
the hash sign command does the same in the other direction
you can prepend a count
three asterisk searches for the third occurrence of the word under the cursor
searching for whold words
if you type the slash the
it will also match there
to only find words that end in the
use
slash the backslash closing angle bracket
the backslash closing angle bracket
item is a special marker that only matches at the end of a word
similarly
backslash opening angle bracket only matches at the beginning of a word
thus to search for the word the only
slash backslash opening angle bracket the backslash closing angle bracket
this does not match there or soothe
notice that the asterisk and hash sign commands use these start of word and end of word makers to only find whole words
you can use G. asterisk and G. hash sign to match partial words
highlighting matches
while editing a program you see a variable called N.R>
you want to check where it's used
you could move the cursor to N.R. and use the asterisk command
and press N. to go along all the matches
there is another way
type this command
colon set H.L. search
if you now search for N.R.
vim will highlight all matches
that is a very good way to see where the variable is used

without the need to type commands
to switch this off
colon set no H.L. search
then you need to switch it on again if you want to use it for the next search command
if you only want to remove the highlighting
use this command
colon no H.L. search
this doesn't reset the option
instead
it disables the highlighting
as soon as you execute a search command
the highlighting will be used again
also for the N. and uppercase N. commands
tuning searches
there are a few options that change how searching works
these are the essential ones
colon set I.N.C. search
this makes vim display the match for the string while yo uare still typing it
use this to check if the right match will be found
then press enter to really jump to that location
or type more to change the search string
colon set no wrap scan
this stops the search at the end of the file
or when you are searching backwards
it stops the search at the start of the file
the wrap scan option is on by default
thus searching wraps around the end of the file
intermezzo
if you like one of the options mentioned before
and set it each time you use vim
you can put the command in your vim startup file
edit the file
as mentioned at not dash compatible
or use this command to find out where it is
colon script names
edit the file
for example with
colon edit tilde slash dot vim R.C.
then add a line with the command to set the opeion
just like you typed it in vim
example
go with the first character in uppercase colon set H.L. search escape
uppercase G. moves to the end of the file
lowercase O. starts a new line
where you type the colon set command
you end insert mode with escape
then write and close the file
double uppercase Z.
if you now start vim again
the H.L. search option will already be set
O. three point nine
simple search patterns
the vim editor uses regular expressions to specify what to search for
regular expressions are an extremely powerful and compact way to specify a search pattern
unfortunately

this powercomes at a price
because regular expressions are a bit tricky to specify
in this section we mention only a few essential ones
more about search patterns and commands can be found in chapter twenty-seven
U.S.R. underscore twenty-seven dot T.X.T.
you can find the full explanation here
pattern
beginning and end of a line
the caret character matches the beginning of a line
on an english dash U.S. keyboard you fid it above the six
the pattern include matches the word include
anywhere on the line
but pattern caret include matches the word include only if it is at the beginning of a line
the dollar sign character matches the end of a line
therefore was dollar sign matches the word was only if it is at the end of a line
let's mark the places where slash the matches in this example line with various X.
using slash the dollar sign
we find this match
and with slash caret the
we find this one
you can try searching with slash caret the dollar sign
it will only match a single line
consisting entirely of the
white space does matter here
thus if a line contains a space after the word
like the space
the pattern will not match
matching any single character
the dot character matches any existing character
for example
the C. dot M. matches a string whose first character is a C.
whose second character is anything
and whose third character is M.
example
matching special characters
if you really want to match a dot
you must avoid its special meaning by a backslash before it
if you search for T.E.R. period
you will find these matches
searching for T.E.R. backslash period only finds the second match
O. three point ten
using marks
when you make a jump to a position with the G. command
vim remembers the position from before this jump
this position is called a mark
to go back where you came from
use this command
theis is a backtick or open single quote character
if you use the same command a second time you will jump back again
that's because the backtick command is a jump itself
and the position from before this jump is remembered
generally
every time you do a command that can move the cursor further than within the same line
this is called a jump
this includes the search commands slash and N.

it doesn't matter how far away the match is
but not the character searches with F.X. and T.X. or the word movements W. and E.
also J. and K. are not considered to be a jump
even when you use a count to make them move the cursor quite a long way away
the double backtick command jumps back and forth
between two points
the control O. command jumps to older positions
hint O. for older
control I. then jumps back to newer positions
hint
for many common keyboard layouts
I. is just next to O.
consider this sequence of commands
thirty-three uppercase G.
slash caret the with the first character in uppercase
control O.
you first jump to line thirty-three
then search for a line that starts with the with the first character in uppercase
then with control O. you jump back to line thirty-three
another control O. takes you back to where you started
if you now use control I.
you jump to line thirty-three again
and to the match for the with another control I.
note
control I. is the same as tab
the colon jumps command gives a list of positions you jumped to
the entry which you used last is marked with a closing angle bracket
named marks
bookmark
vim enables you to place your own marks in the text
the command M.A. marks the place under the cursor as mark A.
you can place twenty-six marks
A. through Z.
in your text
you can't see them
it's just a position that vim remembers
to go to A. mark
use the command backtick mark
where mark is the mark letter
thus to move to the A. mark
backtick A.
the command single quote mark
single quotation mark
or apostrophe
moves you to the beginning of the line containing the mark
this difers from the backtick mark command
which also moves you to the marked column
the marks can be very useful when working on two related parts in a file
suppose you have some text near the start of the file you need to look at
while working on some text near the end of the file
move to the text at the start and place the S.
start
mark there
M.S.
then move to the text you want to work on and put the E. end mark there

M.E.
now you can move around
and when you want to look at the start of the file
you can use thisto jump there
single quote S.
then you can use double single quote to jump back to where you were
or single quote E. to jump to text you were working on at the end
there is nothing special about using S. for start and E. for end
they are just easy to remember
you can use this command to get a list of marks
colon marks
you will notice a few special marks
these include
single quote
the cursor position before doing a jump
double quote
the cursor position when last editing the file
opening bracket
start of the last change
closing bracket
end of the last change
next chapter
U.S.R. underscore O. four dot T.X.T.
making small changes
copyright
see manual dash copyright
vim colon T.W. seventy-eight colon T.S. equals eight colon no E.T. colon F.T. equals help colon no R.L. colon
[usr_03.txt line #623]