

Emacs & D-machine: d source files and dvt

Alex Peyser

September 27, 2011

Contents

1	Introduction & Installation	1
2	D source files	1
2.1	Fontification	2
2.2	Minor modes	3
2.3	Example <code>.emacs</code>	4
2.4	Tabs	4
3	DVT interface	4
3.1	Fontification	5
3.2	Minor modes	5
3.3	Key bindings	6
	References	6

1 Introduction & Installation

The emacs interface to the dvt is composed of a set of lisp files:

```
dm-3.0/  
  site-start.el  
  ap/  
    d-comint-mode.el  
    d-mode.el
```

`site-start.el` should be linked from `/usr/share/emacs/site-lisp/site-start.el` and `ap` should be linked from `/usr/share/emacs/site-lisp/ap` in order to be properly loaded by default. The file `site-start.el` defines the automatic loading of the other files and sets some sensible defaults.

Most emacs commands can either be called from a file in the form *command param1 param2 ...*, such as in your `~/.emacs` file for defaults executed on load, or by typing `Alt-x command` (you will then be prompted for parameters).

The functionality of the d-machine version 3.0, as opposed to the emacs specific interface, is specified in more detail in “D-machine 3.0 Extensions, and interface through DVT,” [Peyser, 2004].

2 D source files

The lisp file `d-mode.el` defines the major mode for editing d code, which is by default defined to be any file of the form `*.[dD]`. Files will also be recognized as d files if they begin with the line:

```
| -*- mode: D; -*-
```

Also, d-mode can be activated on an already loaded file with the interactive command `Alt-x d-mode`.

When a d file is loaded, the file is scanned for carriage returns and if any are found, the user will be queried “Eliminate carriage returns?” Answering ‘y’ will remove the carriage returns and replace them with newlines if necessary.

The user will then be asked whether the file should be immediately saved in its new unix standard format. On the other hand, if 'n' is the response to the first query, the file will be edited as is and newlines will be inserted according to the convention of the file, that is, mac or dos formats.

2.1 Fontification

D mode can fontify text according to emacs standards for programming languages. This is activated in the normal way for emacs, by either calling `global-font-lock` which takes `t` (activates font-lock) or `nil` as a parameter or toggles when called interactively. To change the font-lock-mode for a specific file, call `font-lock-mode` interactively which toggles the font-lock for the current buffer, or define a function for a mode in general as follows in your `.emacs`:

```
(defun my-d-mode-hook ()
  (font-lock-mode t))
(add-hook 'd-mode-hook my-d-mode-hook)
```

Replacing `t` with `nil` will turn off font-lock-mode for d-mode.

The categories for d-mode font-lock are as follows:

font-lock-keyword-face built in operators, such as `list`, `array`...

font-lock-variable-name-face any other legal d name which is not preceded by `/` or `~`

d-mode-oref-face `~name`

font-lock-reference-face `/name`

font-lock-builtin-face a constant such as a number, or `true/false`

font-lock-type-face any of `~[]{}|`

font-lock-comment-face any comment from `|` to newline

font-lock-string-face anything between `(` and `)`, or `<b/w/l/s >`

These can be customized by using the standard emacs font-lock customizer or by setting lines of the following kind in your `.emacs`:

```
(set-face-foreground 'font-lock-keyword-face "red")
(make-face-bold 'font-lock-keyword-face)
```

There is some interference between the comment face and the string face, so escaping unmatched parens in comments is suggested as follows:

```
| Here is an escaped paren \(\ and here is a pair ()
| but the following paren may mess up your font lock (
```

2.2 Minor modes

D-mode has 4 associated minor modes, which are off by default. Each has letter associated with it (such as `p`) which appears on the mode line following `D Machine-` when activated. That letter is also an element of the key sequence which toggles the mode. For example, `C-c C-p` toggles d-mode-magic-parens, and a `p` appears on the mode line when it is active. Modes are:

d-mode-magic-parens(p): typing a mark (such as `{}` `[]`) will re-indent the line to match the preceding line for openers or the matching opener's line for closers.

d-mode-magic-newline(n): hitting `return` will start the cursor at the proper indent position for the next line. Lines will indent the same as the previous line unless it has an unmatched opening mark, in which case it will indent `tab-width` characters further.

d-mode-magic-comments(c): hitting `'|'` will place the comment mark in the same column as the comment mark for the last preceding comment.

d-mode-magic-delete(d): hitting `backspace` (the big delete key on mac keyboards) will delete at least one character, but if the current cursor position is surrounded multiple spaces including newlines & tabs, all preceding spaces will be deleted unless that would join two separate words, in which case all but one will be deleted. For example: (Cursor positon will be marked by `|`)

```

one_two_three|four_five
one_two_three_|four_five

one_two_three_|four_five
one_two_three|four_five

```

2.3 Example .emacs

An example hook for d-mode in ~/.emacs which will turn everything on and set an indent increment of 4 characters:

```

(defun my-d-mode-hook ()
  (setq d-mode-magic-parens t)
  (setq d-mode-magic-newline t)
  (setq d-mode-magic-comment t)
  (setq d-mode-magic-delete t)
  (setq tab-width 2)
  (font-lock-mode t))

(add-hook 'd-mode-hook 'my-d-mode-hook)

```

2.4 Tabs

Tabs will indent the current line, as defined previously. To insert hard spaces regardless of the indent rules, press **Shift-LeftTab** or **Alt-Tab**. Indenting works as per emacs standard, controlled by the commands `tab-always-indent` and other parameters in the ‘indent’ lisp package.

3 DVT interface

The other emacs mode associated with the d-machine is d-comint-mode. This mode is started by calling the command **Alt-x dvt**, which will start a dvt shell running a dvt from `/mnt/Lab1/dm-3.0-arch/dvt`, where arch is g4, g5 or osx depending on the current machine type. The location and name of the dvt to be started can be changed by setting `explicit-dvt-file-name`

in your `.emacs` file, or the environmental variables `EDVT` or `DVT`. The order of precedence is `explicit-dvt-file-name`, `EDVT`, `DVT`, `/mnt/Lab1/dm-3.0/dvt`.

Do it in your `.emacs` as follows:

```
(setq explicit-dvt-file-name "/home/juan/dvt-dir/dvt")
```

By default the buffer will be named `*dvt*`, and if it is already running when `dvt` is called it will switch your current buffer to that buffer, rather than starting a new one. In order to start a `dvt` buffer with a different name, type `C-u Alt-x dvt` instead of `Alt-x dvt`, and you will be prompted for the buffer name.

3.1 Fontification

`DVT` mode shares `d-mode` fontification on the command line typed by the user, but it will not fontify text send by the `dvt` process to emacs. Also, `dvt` mode turns on `ansi-color-for-comint-mode`, converting standard ansi color sequences sent by the `dvt` into actual color. Output from the `dvt` is by default black (has the face `d-comint-mode-highlight-output`), and the input is blue (`comint-highlight-input`), though of course if `font-lock-mode` is on the command line is initially fontified by `d-mode`. Errors are output as red.

3.2 Minor modes

One minor mode for the `dvt` is `d-comint-mode-scream`. It is toggled by `control-!` or `control-1`, and places a `!` in the mode line. When it is activated, any lines typed by the user will have a `!` prepended to it if it does not already have one. This is for sending commands to busy `dnodes`, particularly useful when debugging a `dnode`.

Another minor mode for the `dvt` is `d-comint-mode-redirect-mode`. It is toggled by `control->`, and places a `>` in the mode line. All output from the `dvt` is redirected from the screen and is appended to a file with the same name as the current buffer.

3.3 Key bindings

Key binding for dvt:

- f1** sends the preceding line to the currently selected dmachine (useful for grabbing macros printed by the dvt process to the buffer, and sending it back to the currently selected node as a command)
- f2** sends ! `continue` to the currently selected dmachine and turns off scream mode if active
- f3** sends ! `stop` to the currently selected dmachine and turns off scream mode if active
- f4** sends ! `abort` to the currently selected dmachine, and turns off scream mode if active
- C-!** or **C-1** toggles scream mode
- C-c C-c** clears everything in the buffer from the current cursor position to the beginning of the buffer
- C-c C-a** behaves like **f1**, except the command is wrapped in {} `debug_abort`, for debugging dvt process d code. See `startup_dvt.d` for `debug_abort`'s functionality.
- C->** toggles redirect-mode

For detail regarding selecting a keyboard-owning node or nodes, see Peyser [2004].

References

Alexander Peyser. *D-machine 3.0 Extensions, and interface through DVT*. University of Miami, email: `apeyser@miami.edu`, 2nd edition, June 2004. File: `dvt-3.0.pdf`, original: May 2004.