

Customizing Your EVE Editing Environment

*Academic Computing Support
Tennessee Technological University*

Prerequisite

ACS document 13 or 13a. These documents are available in CH215 and CH313.

Contents

Introduction	2
Customizing EVE during a Current Editing Session	2
Defining Keys	2
Creating Learn Sequences	3
Customizing EVE Automatically Using Startup Files	4
EVE Initialization File	4
EVE Command File	5
EVE Section File	7
For More Information	7

Introduction

EVE (Extensible Versatile Editor) is a text editor based upon the DEC Text Processing Utility (TPU). EVE possesses many powerful features, including the capability of being extensively customized to suit your editing preferences. EVE allows you to redefine default keys, define your own keys, set attributes, create learn sequences and design new TPU procedures to extend your editing environment.

You can make changes to EVE which are in effect for the current editing session only, as well as changes that will apply automatically every time you use the EVE editor.

This document assumes that you are familiar with the basics of using EVE. If you are not familiar with the EVE editor, please consult ACS document 13. *Beginning to Use EVE*.

Customizing EVE during a Current Editing Session

Defining Keys

Many EVE commands may be invoked by either typing that command at the `Command:` prompt or by pressing a key which has been predefined to be equivalent to that command. To see a list of all defined keys, type `help keys` at the `Command:` prompt.

To redefine a key, or define a new one, use the `DEFINE KEY` command. For example, if you are editing two files simultaneously and find yourself moving frequently between the two editing windows, you might want to define `CTRL-N` to be equivalent to the `NEXT WINDOW` command. To do so, you would follow the steps below:

1. At the `Command:` prompt,
Type: define key and press `RETURN`
2. At the `EVE command:` prompt,
Type: next window and press `RETURN`
3. At the `Press the key you want to define:` prompt,
Press: `[CTRL-N]`
The message key defined will appear.

During the remainder of your current editing session, pressing `[CTRL-N]` would be equivalent to pressing `DO`, typing `next window`, and then pressing `RETURN`.

You can assign two different functions to a key by defining a GOLD key with the SET GOLD KEY command. The GOLD key increases the number of possible keys you can define. To use a GOLD key sequence, you press GOLD and then another key. In the following example, the GOLD key is set to PF1 on the keypad, and then the sequence GOLD-C is defined to be equivalent to the command CENTER LINE:

1. At the Command: prompt,
Type: `set gold key pf1`
The message Gold key set appears.
2. At the Command: prompt,
Type: `define key= gold-c center line`

You can use most CTRL and PF key sequences, and function keys F7-F20 for your key definitions. However, redefining any of the following is not recommended: CTRL-B, CTRL-C, CTRL-I, CTRL-O, CTRL-R, CTRL-S, CTRL-T, CTRL-U, CTRL-V, CTRL-X, CTRL-Y, CTRL-Z, or the HELP and DELETE keys.

For more information about defining keys, type `help define key` or `help set gold key` at the Command: prompt.

Creating Learn Sequences

A learn sequence is EVE's name for what is commonly called a macro: a single key which, when pressed, executes a series of keystrokes (such as inserting text, moving the cursor and invoking EVE commands). To define a learn sequence, use the LEARN command:

1. At the Command: prompt,
Type: `learn`
2. *Type the keystrokes you want the defined key to repeat. You can insert text, give EVE commands or use any keys that are defined.*
3. *When you're done, press: [CTRL-R]*
4. *Press the key you want to define (for example, [CTRL-N]).*
The message key sequence remembered will appear.

For the remainder of the current editing session, pressing [CTRL-N] would be equivalent to carrying out the series of keystrokes listed in step two above.

For more information about creating learn sequences, type `help learn` at the Command: prompt.

Customizing EVE Automatically Using Startup Files

EVE Initialization File

You can predefine the characteristics of all your editing sessions by creating an EVE initialization file, which is a text file containing EVE commands. Commands in an initialization file can also be typed at the EVE Command: prompt, which is a good way to try them out before putting them in a startup file.

Here is a sample EVE initialization file that you could create (using EVE, of course!):

```
! set some attributes
  set left margin 8
  set right margin 72
  set tabs every 4
! key definitions
  set gold key pf1
  define key= ctrl-d two windows
  define key= ctrl-n next window
  define key= ctrl-f find
  define key= gold-c center line
  define key= gold-e exit
  define key= gold-q quit
```

By default, EVE looks for a file named EVE\$INIT.EVE in your current directory to use as an initialization file. If you will be working in different directories or want to use the same initialization file for editing mail messages, you will want to define the logical name EVE\$INIT to point to the same initialization file. To do so, place the following line in a file named LOGIN.COM in your top-level directory:

```
$ define EVE$INIT sys$login:eve$init.eve
```

If you want your EVE\$INIT.EVE file to reside in a subdirectory rather than your top-level directory, you will instead need to specify that in your LOGIN.COM file. For example:

```
$ define EVE$INIT user:[jqu1234.bin]eve$init.eve
```

EVE Command File

A more advanced startup file is an EVE command file. Command files execute more quickly than initialization files and offer more sophisticated editing tools. An EVE command file contains programs written in the DECTPU language. DECTPU (DEC Text Processing Utility) is the programmable text processor upon which EVE is built.

Here is a sample EVE command file that you can create (or copy):

```
!Sample EVE command file; to obtain a copy of
!this file, type the following at the $ prompt:
!      copy user:[acs.eve]tpu$command.tpu *

!Quote function for MAIL;
!To execute while editing a mail message,
!type: MAILQUOTE at the Command: prompt.
procedure eve_mailquote
  position (buffer_begin);
  loop
    exitif (mark(none) >= end_of (current_buffer));
    if current_character <> ">"
      then
        copy_text ("> ");
      else
        copy_text (">");
      endif;
    move_vertical (1);
    position (line_begin);
  endloop;
endprocedure

!Transpose character and previous character;
!To execute while editing a file,
!type: SWAP at the Command: prompt.
procedure eve_swap;
  local swap_this;
  swap_this := erase_character (1);
  move_horizontal (-1);
  eve$insert_text (swap_this);
  return (true);
endprocedure;
```

```

!When editing a c program, set tab stops every four spaces;
!Otherwise, set the tab stops every eight spaces.
!This will be executed automatically when EVE is invoked
!by including the command eve_setmargins at the bottom
!of this command file.
procedure eve_setmargins;
x := get_info (current_buffer,"output_file");
str1 := file_parse (x,""," ",type);
if str1 = ".C"
then
    set (tab_stops,current_buffer,4);
else
    set (tab_stops,current_buffer,8);
endif;
endprocedure;

!This defines the GOLD key to be PF1 and
!then GOLD-H = "include signature.txt";
!Create signature.txt separately. To include your signature
!file at the bottom of a mail message you are editing, move to
!the bottom of the message and press GOLD-H.
set (shift_key,pf1);
define_key ("eve_include_file('signature.txt')",key_name('h',shift_key));

eve_setmargins;

```

Once you have created an EVE command file, you can define a logical name to point to it by adding a line such as the following to your LOGIN.COM:

```
$ define TPU$COMMAND sys$login:tpu$command.tpu
```

When you invoke EVE, your command file will automatically be compiled. To execute any of the procedures in it, type the name of the procedure, without the eve_ prefix, at EVE's Command: prompt. For example, to execute the procedure eve_mailquote, you would press DO, type mailquote and press RETURN. You can also define keys to execute procedures in your EVE command file, either by defining a key as shown above, or by including a key definition in your EVE initialization file. For example:

```
define key= gold-r mailquote
```

If you want to learn more about programming in DECTPU, consult the DEC Text Processing Utility Manual in CH313A. For information about TPU built-in procedures that you can use, type `HELP TPU` at EVE's `Command:` prompt. If you would like to see some more advanced examples, the TPU source code for EVE is located in `sys$examples:eve$.tpu`.

EVE Section File

A third type of EVE startup file is a section file. A section file contains compiled DECTPU programs and statements, and executes slightly faster at startup than initialization files and command files. Since EVE section files are extremely large, and the startup time increase is marginal, we suggest that you use initialization and command files to customize EVE instead. For more information about section files, consult the EVE manual or type `help section files` at EVE's `Command:` prompt.

For More Information

For more information about using EVE, contact Academic Computing Support.