

# Using nu/TPU on UNIX

Last revised: 1/28/00

nu/TPU is a general-purpose text editor available on UNIX that simulates the TPU editor available on VMS operating systems. By default nu/TPU emulates the EDT keypads but it can also emulate the EVE keypads. Unless otherwise noted, all the references to keys in this handout assume you are using the default EDT keypad emulation.

You can use nu/TPU to create and edit new files or to edit existing files. This document will show you how to get started using nu/TPU and teach you some of its basic features.

## A Note about Terminal Types

How well nu/TPU emulates the EDT or EVE keypad depends on the type of terminal you are working at. NCD X-terminals like those in the Public Terminal Rooms perform very well with nu/TPU and work "as advertised" for the most part.

Performance from PC's depends on the type of terminal emulation software you are using. VT200 provided the best emulation. We tested nu/TPU with WiscWorld's terminal emulation program called SecureCRT. Check your session preferences for SecureCRT and make sure VT220 emulation is checked and that the keyboard mapping is set to VT220 as well. A few keys on the keypads don't work as advertised but these are noted in this handout.

nu/TPU performed very poorly using VT100 emulation. Performance was also unsatisfactory from Xwin32 terminal emulation. The numeric keypad did not work at all.

## Commands in nu/TPU

Commands in nu/TPU are entered one of the three following ways:

- by using keys that are predefined by nu/TPU. The keys predefined by default on VT200-series terminals include the numeric keypad, the edit keypad (located between the main keyboard keys and the numeric keypad), the cursor keypad (located below the edit keypad), and certain function keys. The section, *Using the Keypads*, describes how to determine the keypad mapping for commands.
- by entering a sequence of keystrokes starting with the Control key followed by another key. The section, *Using the Keypads*, describes how to determine which commands are mapped to Control key sequences.
- by entering commands from the command line. nu/TPU commands can always be entered from the command line. To display the command line, press <Do>. If you do not have a Do key, press <Ctrl-B>.

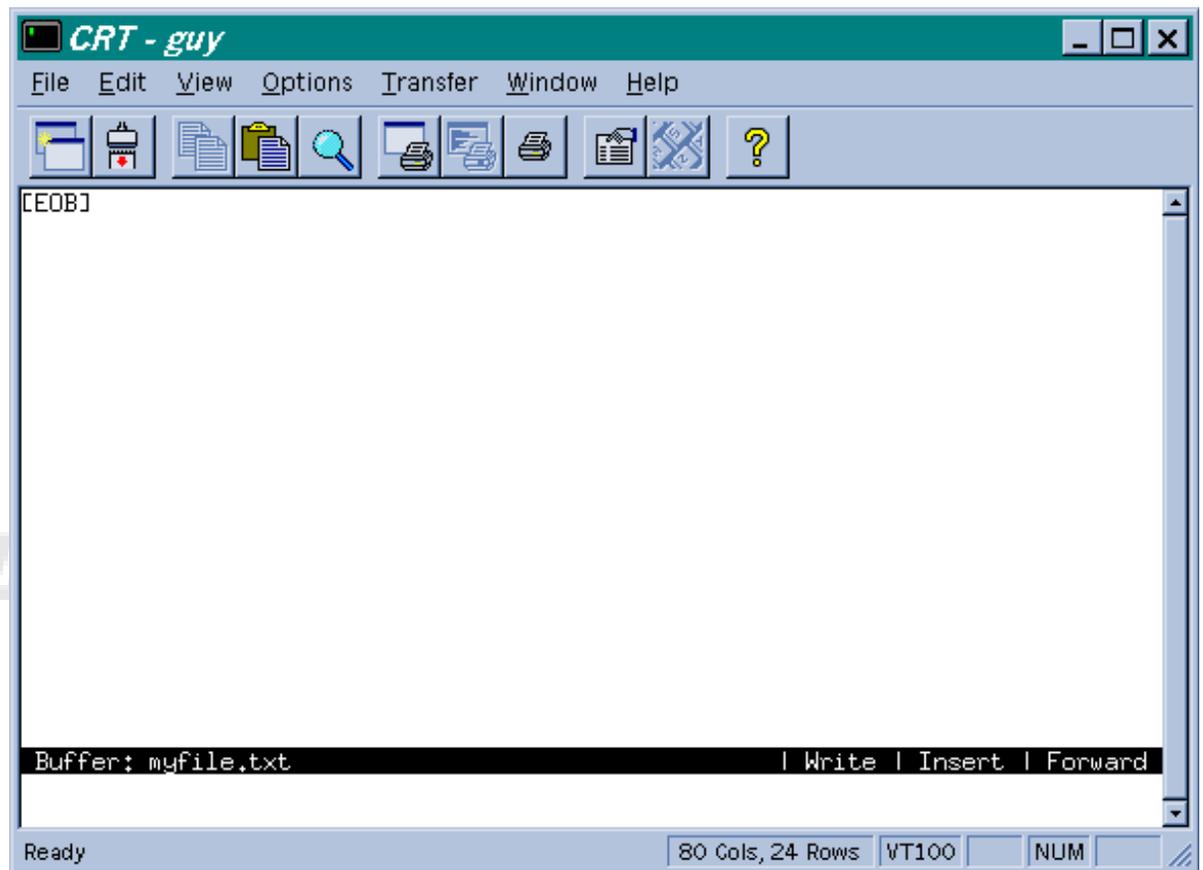
## Entering and Exiting the Editor

You can start an nu/TPU editing session either by creating and editing a new file or by

editing an existing file. To begin an editing session, enter the command `tpu` followed by the name of the new file you want to create or the existing file you want to edit. (Type `tpu -h` for a list of options that can be used with the `tpu` command.) For example, to invoke nu/TPU and create a new file named `myfile.txt`, type:

```
> tpu myfile.txt
```

You will see a mostly blank screen which looks something like this:



If you specify an existing file on the nu/TPU command line, nu/TPU displays the text of the file on the screen. When a file is loaded into nu/TPU, the location of the file is referred to as a **buffer**. This is a temporary holding area for your file.

[EOB] marks the end of an nu/TPU buffer and the end of your file. It is visible only on the screen and does not become part of your file. When you add lines of text, [EOB] moves down.

A highlighted **status line** appears at the bottom of the nu/TPU window and provides information about the buffer you are viewing in the window. The status line shows the buffer name, editing status (write-able or read-only), current typing mode (insert or overstrike), and current search direction (forward or reverse).

A **message area** lies beneath the highlighted status line. During the editing session, nu/TPU displays messages in this area.

Once you are in the editor, you can begin typing.

## Ending an editing session

When you are finished editing the file, you can either save your changes (EXIT) or discard them (QUIT).

To end the editing session and *save* your edited text, use the EXIT command. You can execute the EXIT command by pressing the <F10> or by pressing <Ctrl-Z>. If neither of these methods work, press <Ctrl-B> and then type `exit` on the command line.

If you have modified the current buffer, nu/TPU creates a new version of the file with the same file name as the original version. A backup version is also created and has an added file extension of `.%1`. For example, if you use the EXIT command after modifying a file named `myfile.txt`, the saved file is named `myfile.txt` and the backup version is named `myfile.txt.%1`. To create more than one backup version, you should include the BACKUP command in a nu/TPU defaults file, discussed later in this document.

To end the editing session *without* saving the changes that you made, use the QUIT command. With the QUIT command, the editing session ends and any changes that you made to the file are ignored. Any existing versions of the files remain unchanged. To execute the QUIT command, type `quit` from the command line. (To display the command line, press <Do>. If you do not have a DO key, press <Ctrl-B>.)

You will see a message like this on the screen:

```
Buffer modifications will not be saved, continue quitting [Yes]?
```

Type `y` and press <Return> if you want to quit without saving the changes. If you change your mind, type `n` and press <Return>.

Note: If you find yourself in a situation where you are "stuck" in nu/TPU and cannot exit the program, you can abort the program by entering <Ctrl-C> twice. Use <Ctrl-C> sparingly because aborting your session does not allow you to save your work.

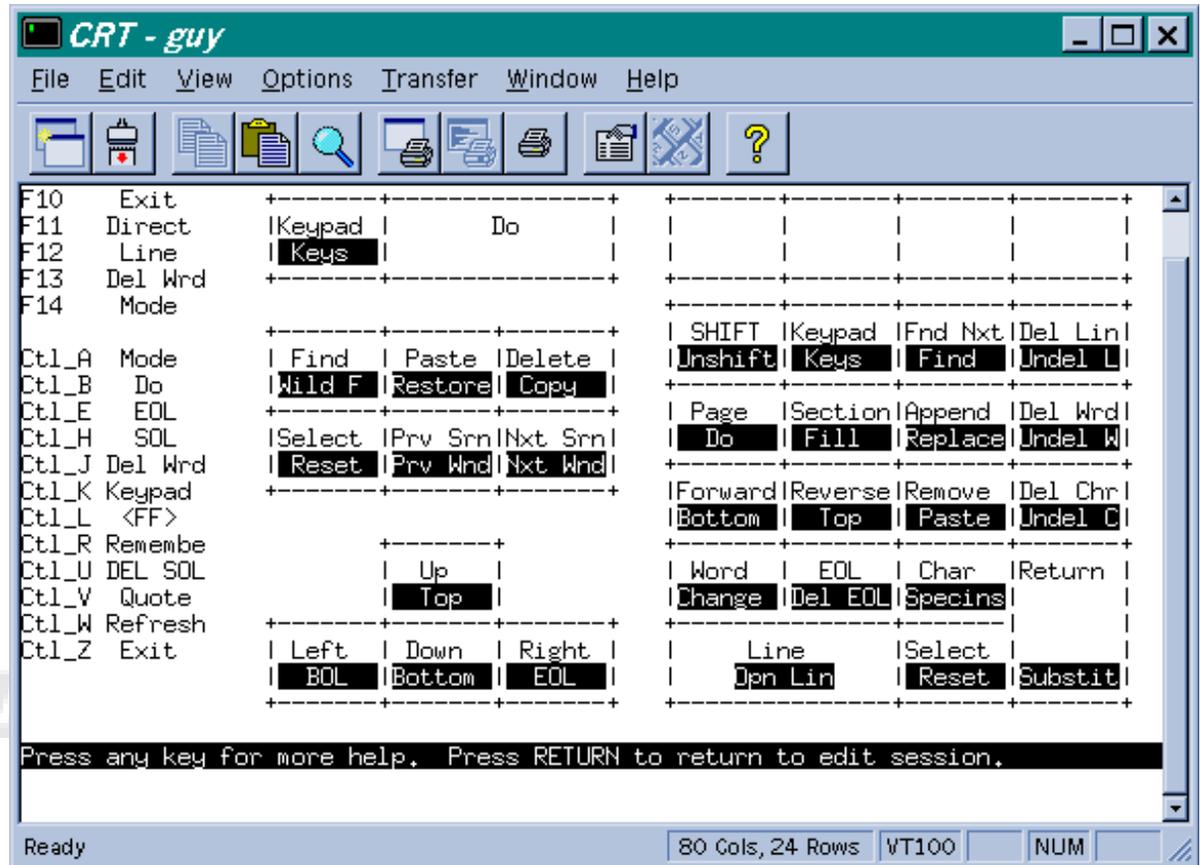
## Using the Keypads

Some keys described in this document are only available on NCD X-terminals or PCs running WiscWorld's Telnet program called SecureCRT where you can emulate a VT200-series terminal. If you are using a different type of terminal emulator, your keys may function differently and require some experimentation.

nu/TPU defines some keys by default. The predefined keys on the VT200-series terminals include the numeric keypad, the edit keypad (located between the main keyboard keys and the numeric keypad), the cursor keypad (located below the edit keypad), certain function keys, and certain control key sequences.

By default, your keypads and control key sequences are defined to emulate EDT. To get a display of your keyboard mapping, press <HELP> if you are using an X-terminal or

press <Ctrl-K> from other terminal types. Below is the default keyboard mapping (EDT emulation) display for terminals emulating a VT200-series:



With this window displayed, you can press any key for more information on that specific key. To exit this HELP screen, press <Return>.

Most keys have two definitions preassigned. To use the key for the definition that is displayed first on the key, just press the key. For example, to move your cursor down one line, press the **Down** key (down arrow key on the cursor keypad).

To use the key for the definition that is displayed in reverse video on the screen above, you must first press the key corresponding to the key labeled **SHIFT** on the numeric keypad in the screen above (The Num Lock key on a PC keyboard and <PF1> on X-terminals). Then press the key corresponding to the command you want to execute. The **SHIFT** key is also known as the **GOLD** key. For example, to move to the Bottom of your document, first press **SHIFT** on the numeric keypad and then press **Bottom** (down arrow key) on the cursor keypad.

If you are working from a PC (rather than an X-terminal), you will notice that you do not have all the keys pictured in the diagram above. In particular, PC keyboards do not have the two keys located above the Edit keypad (**KeyPad/Keys** and **Do**) and they do not

have the **Del Wrd/Und Wrd** key on the last column of the numeric keyboard (because there are three keys rather than four on this column). The **Keypad/Keys** key displays the keypad mappings and you can press <Ctrl-K> instead. The **Do** key gets you to the command line and you can press <Ctrl-B> instead. The **Del Wrd** key deletes the word the cursor is on and you can press <Ctrl-J> instead. Read the *Defining Keys* section to see how to define a key for **Und Word**.

## Emulating the EVE Keypad

To emulate the EVE keypad, type `set keypad eve` from the command line. (To display the command line, press <Do>. If you do not have a DO key, press <Ctrl-B>.)

To get a display of the EVE keyboard mapping, press <HELP> if you are using an X-terminal or press <Ctrl-K> from other terminal types. Below is the EVE keyboard mapping display for terminals emulating a VT200-series:

```

CRT - guy
File Edit View Options Transfer Window Help
[Icons]
F10  exit
F11  Forward  |help kel      do
F12  move_by   |help kel
F13  erase_w
F14  Insert
      +-----+
Ctl_A Insert  | find |insert_|delete |
Ctl_B recall  |find_ne|restore|copy
Ctl_E end_of_
Ctl_H start_o |select|previous|next_sc|
Ctl_J erase_w |reset |previous|next_wi|
Ctl_K Keypad
Ctl_L insert_
Ctl_R
      +-----+
Ctl_U erase_s |move_up|
Ctl_V quote   |top
Ctl_W refresh
Ctl_Z exit    |move_le|move_do|move_ri|
              |start_o|bottom |end_of_|
      +-----+
Press any key for more help. Press RETURN to return to edit session.
EVE keypad active.
Ready 80 Cols, 24 Rows VT100 NUM

```

Note: The EVE keypad emulation does not work well from many PC keyboards. If keys do not work as defined in the screen above, you will need to reset your keypad to EDT.

## Entering Commands at the Command Prompt

In addition to using defined keys to perform commands, you can enter any nu/TPU command at the `Command:` prompt. To execute nu/TPU commands from the `Command:`

prompt, do the following:

1. Press **DO** (or <Ctrl-B>). nu/TPU displays the `Command:` prompt in the command window beneath the highlighted status line.
2. Type the nu/TPU command you want to perform. The following example shows how to enter the EXIT command at the prompt:

```
Command:  EXIT
```

3. Press <Return> to execute the command.

For a complete list of all the nu/TPU commands, press **DO** and enter `help` on the command line. This display groups commands by functionality. To get more information on a specific command, enter its name.

The following tips can help you save keystrokes when typing nu/TPU commands:

- ! To recall the last nu/TPU command you entered, press the up arrow from the command line. Continue pressing the up arrow until the command you want appears at the prompt, then press <Return> to execute it.
- ! To abbreviate nu/TPU command names, use the first letters of each command term, making sure to use enough letters to uniquely identify the command. A maximum of four characters for each word of the command is required. For example, if you wanted to change to overstrike mode, you could type `over` at the `Command:` prompt. To change to insert mode, you could type `INS MODE` -- just `INSERT` is not enough to uniquely identify the command.
- ! To repeat an nu/TPU command or keystroke a specified number of times, use the `REPEAT` command. Enter the `REPEAT` command and the number of times it is to be repeated. nu/TPU repeats the next character or command you enter the specified number of items. For example, the following commands delete 4 lines:

```
Command: REPEAT 4
Will repeat next command 4 times.
Command: ERASE LINE
```

- ! To repeat the last command entered, press **DO** twice.

## Defining Keys

As you saw above, nu/TPU by default has many predefined keys. However, if there is a command you repeatedly use which is not defined, you can use the `DEFINE KEY` command to add a function to your keyboard. The following example assigns the nu/TPU function `RESTORE WORD` to a key of your choice for the duration of the editing session.

1. Press **DO** (or <Ctrl-B>).

2. At the `Command:` prompt, type `define key`
3. At the `nu/TPU` command: prompt, type `restore word`
4. At the `Press the key that you want to define:` prompt, press the key which you want to perform the `Move By Word` function.

If you want this function to remain in place for future editing sessions, you must define the key in your `nu/TPU` initialization file, described later in this document.

## Moving Around The Screen

`nu/TPU` provides a variety of cursor movement commands:

BOTTOM	MARK	MOVE RIGHT	SET CURSOR FREE
CHANGE DIRECTION	MOVE BY LINE	MOVE UP	SET SCROLL MARGINS
END OF LINE	MOVE BY PAGE	NEXT SCREEN	START OF LINE
FORWARD	MOVE BY WORD	PREVIOUS SCREEN TOP	
GO TO	MOVE DOWN	REVERSE	WHAT LINE
LINE	MOVE LEFT	SET CURSOR BOUND	

To find out the function of each of these commands, type `HELP` followed by the name of the command from the command line. For example:

```
Command: HELP BOTTOM
```

You can always execute any of these commands from the command line, But, many of these commands are bound to keys on your keyboard. A quick mapping of the cursor keys is provided when you press `<HELP>` from an X-terminal or `<Ctrl-K>` from other terminal types.

## Editing Text

Once you know how to invoke the `nu/TPU` editor, move your cursor around the screen, and how to enter commands, you can use `nu/TPU` commands to help you create and edit files. Below are the commands `nu/TPU` provides for editing text:

CHANGE MODE	ERASE WORD	RESTORE CHARACTER
COPY	INSERT HERE	RESTORE LINE
CUT	INSERT MODE	RESTORE SELECTION
DELETE	OVERSTRIKE MODE	RESTORE SENTENCE
ERASE CHARACTER	PASTE	RESTORE WORD
ERASE LINE	QUOTE	SELECT
ERASE PREVIOUS WORD	REMOVE	SELECT ALL
ERASE START OF LINE	RESTORE	STORE TEXT

To find out the function of each of these commands, type `HELP` followed by the name of the command from the command line.

You can always execute any of these commands from the command line, But, many of

these commands are bound to keys on your keyboard. A quick mapping of the cursor keys is provided when you press <HELP> from an X-terminal or <Ctrl-K> from other terminal types.

Before you begin typing text, look at the highlighted status line to check whether the buffer is in insert or overstrike mode. If the buffer is in insert mode, text is inserted at the cursor position, and text that already appears in the file moves to accommodate your insertions. If the buffer is in overstrike mode, text that you type at the keyboard is inserted at the cursor position, and the text that already appears in the file is overwritten as the cursor moves through it. Press <F14> from X-terminals or <Ctrl-A> from PCs to change from one mode to the other.

## Moving Text

nu/TPU lets you cut and paste text. Follow these steps:

1. Select the text you want to move, like this:
  - Position the cursor at the beginning of the text.
  - Press **SELECT**.
  - Move the cursor to the end of the text.
2. Remove the text by pressing **REMOVE**.
3. Move the cursor to the new location.
4. Insert the text by pressing **INSERT HERE**.

Note that the selected text is displayed in reverse video. If you change your mind when selecting text you can unselect by pressing **SELECT** again. You can also insert the text more than once by pressing **INSERT HERE** again and again.

If your keyboard does not include these keys, you can execute all these commands from the Command: prompt.

## Finding Text

To locate specific text in the current buffer, press **FIND** or use the FIND command. Then type the text you want to locate, which is called the "search string". For example, the following commands search for the string *This is* from the current cursor position forward:

```
Command: FIND
Search For: This is
```

If the string is found, nu/TPU moves the cursor to the beginning of the specified string.

If the search string contains all lowercase letters, nu/TPU disregards the case of the letters and locates any occurrence of the string. Thus, the search string *this* matches *this*, *THis*, and *ThiS*. If the search string contains one or more uppercase letters, nu/TPU finds only the occurrences of the string in which the case of each letter is exactly the same. Therefore, the only match for the search string *tHe is* is *tHe*.

The direction within the buffer that is searched can be changed. Recall that the current search direction is displayed in the status line of the window. To change the direction, press either **Forward** or **Reverse** (4 and 5 on the numeric keypad).

To find the next occurrence of the search string, press **FIND** twice or enter the FIND NEXT command.

## Replacing Text

The REPLACE command lets you replace a text string in the current buffer with another text string.

Like FIND, REPLACE is case sensitive. If the old string has any uppercase letters, nu/TPU searches for exact case matches. If the old string is all lowercase, nu/TPU searches for any occurrence of the string regardless of its case. If the new string has any uppercase letters, nu/TPU replaces the string exactly. If the old and new strings are all lowercase, nu/TPU replaces the string according to the following rules:

- ! A capitalized version of the old string (first letter uppercase, others lowercase) is replaced by a capitalized version of the new string.
- ! An all uppercase version of the old string is replaced by an all uppercase version of the new string.
- ! Otherwise, the old string is replaced by an all-lowercase version of the new string.

Follow these steps to replace text:

1. Type `replace` at the command line.
2. At the `Old text:` prompt, type in the text you want to replace and then press `<Return>`.
3. At the `New text:` prompt, type in the replacement text and press `<Return>`.
4. Enter the appropriate response to the prompt:

Replace? Type Yes, No, All, Last, or Quit [Yes]:

## Session Recovery

Nu/TPU's journal facility can recover your work if your editing session is interrupted unexpectedly. A system interruption might occur due to:

- ! Loss of electrical power
- ! A break in the telephone link if you dialed into the system
- ! A system crash

As you edit, nu/TPU automatically stores every keystroke you make in a file called a journal file. The journal file contains all of the text and editing commands that you entered during the editing session. After the interruption, you can restore the changes made to your file to the point just before the interruption. The journal file has the same name as the file that you were editing, except the file type is .TJL. To recover a journal file, enter the following command at the UNIX prompt:

```
> tpu -recoverfilename
```

Where *filename* is the name of the file you were editing, NOT the journal file. Note that when nu/TPU recovers a session, it inserts all the keystrokes which were stored in the journal file. When the recover is complete, you can continue editing the file. Sometimes the last few keystrokes in an editing session are not stored in the journal file, so be sure to check for this before continuing your editing session.

## Buffers and Windows

You may have many buffers open in any one editing session. That is, you can edit many files at once. You can also have many windows on your screen simultaneously. Different windows can display only the same buffer or different buffers. You may cut and paste between windows and buffers.

To create a buffer, type the following command:

```
Command: GET FILE filename
```

This creates another buffer with the name specified by the file specification.

To select among the buffers already created, type the following command:

```
Command: SHOW BUFFERS
```

A window will appear listing all the buffers. By using the up and down arrow keys along with the select and remove keys, you can delete a buffer or select a buffer for editing. You can also use the command `NEXT BUFFER` to switch between buffers.

To create more than one window on the screen, type the command:

```
Command: TWO WINDOWS
```

or

```
Command: SPLIT WINDOW integer
```

(`SPLIT WINDOW 2` is the same as `TWO WINDOWS`)

These commands will divide the current window into the specified number of windows. The maximum number of windows you can have current depends on the type of terminal you are using.

You can move among your windows by using **SHIFT** (The Num Lock key on a PC

keyboard and <PF1> on X-terminals) in combination with **PRV SCR** and **NEXT SCR**. You can change the size of your windows with the **SHRINK WINDOW** and **ENLARGE WINDOW** commands. Each uses an argument of an integer (e.g. **SHRINK WINDOW 5**) to shrink or enlarge the current window the specified number of lines. The command **ONE WINDOW** will delete all but the currently active window, returning you to a single window environment.

## Learn sequences

The **LEARN** command records a sequence of keystrokes and assigns them to a single key. The sequence can consist of commands, text, or both. (In some text editors this is called a "macro.") To create a **LEARN** sequence, follow these steps:

1. Type **LEARN** at the Command: prompt. You will see the message `Press keystrokes to be learned. Press CTRL/R to remember these keystrokes.`
2. Type the keystrokes you want learned. You can type text, commands, or both -- including pressing previously defined keys. However, mouse operations are NOT learned.
3. To end or "remember" the learn sequence, press <Ctrl-R>. You will see the message `Press the key you want to use to do what was just learned:`
4. Press the key you want to use for the learn sequence. You can press:
  - ! A function key ..... such as <F20> or <PF4>
  - ! A keypad key ..... such as Keypad 9
  - ! A **SHIFT (GOLD)** combination ... such as **GOLD A**
  - ! A control key ..... such as <Ctrl-N>
  - ! <Return> ..... to cancel the definition

The learn sequence remains assigned to the key throughout your editing session or until you redefine the key.

Note: Learn sequences record keystrokes, rather than the key definitions at the time the sequence was created. Therefore, when you "replay" a learn sequence, you should be sure that any keys used in the learn sequence are still defined appropriately and also that any settings that affect the sequence are correct (such as bound or free cursor, margins, and tab stops, pending delete, and so on). Otherwise, the learn sequence may fail or may not work as expected.

## Customizing nu/TPU

nu/TPU provides many tools that allow for extensive customization of the editor. nu/TPU is customized by creating certain files that nu/TPU looks for when you invoke the editor. The manual describes each of these files in great detail. In this section, two of the most

commonly used files for customization are explained: the initialization file and the defaults file.

## Initialization File

If you define some keys frequently, you can avoid having to define them every time you enter the editor by creating an initialization file. Every time you call the nu/TPU editor, it searches your home directory for a file called `tpu_init.tpu`. If the file exists, the editor executes all the nu/TPU commands in the file.

The initialization file should contain nu/TPU commands, including key definitions and editing values, such as margins and tabs. For example:

```
!define some keys
define key=F7 move by word
define key=CTRL_B_KEY erase word
!Disallow cursor from moving past the end of a line
set cursor bound
!the next command defines the GOLD (SHIFT) key
set gold key F5
```

The example above shows a simple initialization file. You may want to create a file called `tpu_init.tpu` in your home directory and include some or all of these commands in the file. (You should only include the SET KEYPAD EVE command if you will want to use the EVE keypad.) If you do this, these values will always be set for you when you access nu/TPU.

SOCIAL SCIENCE COMPUTING COOPERATIVE

Each command must be on a separate line -- no continuations. If a command is incomplete, nu/TPU prompts you for any required information before continuing to execute the initialization file.

You can put many commands in the nu/TPU initialization file, but the longer the file, the longer it will take for you to access nu/TPU. In general, you should only include frequently-used commands. You can always define other keys and editing values for the current editing session.

Comments in the initialization file must be on lines separate from commands and must begin with an exclamation point (!).

## Defaults File

The defaults file allows you to override nu/TPU's environment settings. Environment settings include options you would normally type on the `tpu` command line and operating system options like the number of backup files to save, keyboard type, width of your window, etc.

To include a defaults file, you need to create a file called `nu_tpu.def` in your home directory. Below is the contents of a simple defaults file:

```
# Refer to the Reference manual for the complete
```

```
# list items defined in a defaults file.  
#  
# Width of the window:  
WIDTH:72:  
#  
#Save two backup files:  
BACKUPS:2:
```

The defaults file's syntax provides for a single keyword setting per line. The keyword must start in the first column and be terminated by a colon. Immediately after the colon is the new value for the keyword. Space and tabs are not allowed before or after the colon separating the keyword and its value. The text for the new value can be terminated with a colon or a space character. Pound signs, #, indicate comments.

## nu/TPU documentation

nu/TPU is documented in the following manuals available for short term loan from the CDE Print Library in Social Science 4457:

*nu/TPU v5.0 Reference Manual*

*nu/TPU Lesson Guide*

SOCIAL SCIENCE COMPUTING COOPERATIVE

