

Appendix K

Keyboard Shortcuts of NUSS

This appendix details alternative *keyboard shortcuts* that can be used to execute most NUSS functions — when the *mouse* is *not* the preferred method of program manipulation — or when keyboard data entry is occasionally required anyway. Though the mouse can be a very useful tool, when you must make a data entry via the keyboard, you are required to put down the mouse and move to the keyboard. Once your hands have moved to the keyboard, it is valuable to know other efficient ways to continue your work without necessarily having to move your hands some distance to pick up the mouse again.

The shortcuts described here are particularly useful when navigating (or filling in user-modifiable data) on *forms* — which are the principal user-visible building blocks used by NUSS. A typical form has several data entry fields (*text boxes*), selectable options (*radio-buttons* and *check-boxes*), and other controls (*buttons*). The mouse navigation methods for moving among these elements is generally obvious (once you are Windows oriented). The alternate keyboard navigation methods are less obvious. Whenever possible, NUSS provides keyboard shortcuts per standard Windows conventions.

To address any form by keyboard (or even the mouse) that form (or window) must first have the Windows *focus*. *Form focus* is obtained by making a form the one form (or window) on the screen with its **Title Bar** highlighted. This generally requires clicking the Title Bar (or any clear area) of desired form with the mouse. Although the <Alt>+<Esc> combination can navigate from application-to-application on the screen, it does not appear to work for moving from form-to-form within a multi-form application (like NUSS) — which may always require *mouse* selection by clicking the form's *title bar* or its *button* on the Windows *task bar*.

Once a particular form gets the Window's *form focus*, then any of the individual controls on that form can also be given *control focus*. This *control focus* may be observed by the appearance of a *focus cursor* of some kind on the *first* control or data field of that form. If it appears in a *text box*, it is the *vertical line text entry cursor* common to word-processing applications. If it appears on a *button* or *control label*, it is a *dotted-line* surrounding the label's text. Every such control or data field on the form is linked by a programmable function (known to the programmer as its Tab index sequence). One control is always programmed to begin the sequence (i.e., given tab index = 1).

The <Tab> key (or <Shift>+<Tab> key) allows any form's *focus cursor* to move forward (or backward) across the form's fields and controls, per its defined Tab sequence.

The <**arrow**> keys (known separately as <**Left**> or <**Right**> or <**Up**> or <**Down**> keys) can also be used to move this focus. If you need to move in a more *random* motion across a form, the various controls may have *labels* with an *underlined character* in them. When such labels appear you may move directly to that control by the <Alt>+<*shortcutkey*> key combination, where <*shortcutkey*> is the *underlined character*. If more than one control on a form is assigned the same <*shortcutkey*> one is selected first, but pressing it again will select the next.

Once you have chosen (highlighted by vertical or dotted cursor) a particular control or field, and want to execute it, the <**Space**> key is usually the closest equivalent of a mouse click of that control or field. However, when several *radio buttons* provide a choice inside a *frame* of controls, then the various <**arrow**> keys can make the correct selection and execute it in one move. For example, the <**Space**> key alternately *checks* then *unchecks* a *check box* control that is currently highlighted if you press it more than once. The <**Space**> key simply “presses” the currently highlighted *button* control that has *control focus*.

Menu selections have their own rules, and may have both <*hotkeys*> and <*shortcutkeys*> associated with them. The <*shortcutkey*> is again a *single underlined character* in the *label* of the selection, that randomly selects (highlights) it. Then <**Enter**> is required to execute it. However, a <*hotkey*> is usually documented (in parentheses) at the end of a menu label. It usually is a unique control key: <Ctrl>+<*otherkey*> combination that both *selects* and *executes* the function in one operation. See the last paragraph, below, for how to navigate menu selections once a menu section is visible.

Every form can be programmed so that the <**Esc**> key and <**Enter**> key immediately executes a particular assigned control when pressed. Many NUSS forms are programmed to press that form’s [X] or [Cancel] button when the <**Esc**> key is pressed. Many NUSS forms are also programmed to press the [Exit] or [OK] button when <**Enter**> is pressed.

Currently, there seems to be no equivalent *standard* Windows keyboard function for popping-up a *context menu* in a field, form, or window. You must *right-click* your mouse to perform this convenient function. However, NUSS has programmed the <**Enter**> key to pop up its *single-module context menu* on the **Nodes Map** of the **Network Status** form, whenever the desired module’s *serial # icon is already highlighted*. Pressing the <**Enter**> key a *second* time erases the context menu (with a beep). You may also move from one *module’s icon* to another using the <**arrow**> keys. Unfortunately, you must still use the mouse to highlight the *first* module icon on this form.

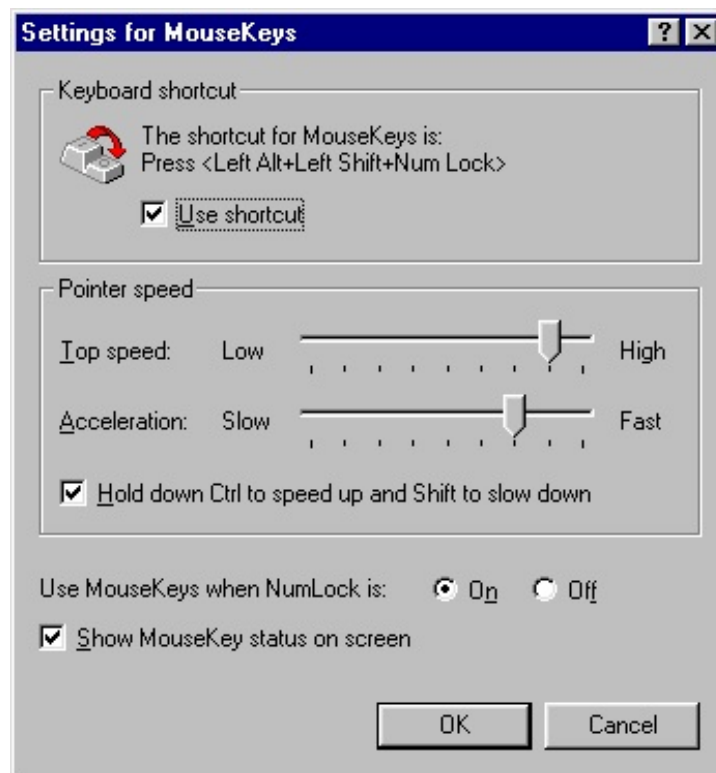
Once any menu section is visible, you may use the <**arrow**> keys to find the desired menu selection, and then press <**Enter**> to execute the final menu choice. The <**Up**> and

<**Down**> arrow keys traverse all the displayed selections on the menu. The <**Right**> arrow key pops-up any submenu available on the highlighted main menu function. The <**Up**> and <**Down**> arrow keys traverse the submenu selections as well. The <**Right**> and <**Left**> arrow keys may also be used to move between all the menu items on any form's menu bar.

Using MouseKeys

If you are a fanatic about never taking your hands off the keyboard, or if you ever find yourself without a functioning mouse on your PC, you may be interested in learning how to use the **MouseKeys Accessibility** feature of Windows.

To enable **MouseKeys** (assuming it is installed) activate the **My Computer** form, Click the **Control Panel** icon, then the **Accessibility Options** icon, and finally the **Mouse** tab. Check the **MouseKeys** check box (if not already), and then click its **[Settings...]** button. The following form appears:



You may later return to this form to adjust things to your taste, but for now use the options shown, press **[OK]**, and then press **[Apply]** and **[OK]** on the “outer” form.

To enable (and later disable) the **MouseKeys** feature use the shortcut key combination <Alt>+<Shift>+<Num Lock>. Thereafter, pressing <Num Lock> alone toggles the *numeric keypad* between its *normal* uses and its **MouseKeys** use. When it is enabled, a small “mouse” icon appears in the system tray (next to the Windows Time). This icon is very useful in indicating the mode of the ‘5’ keypad key, that has many uses.

The following table lists the purpose of each key on the numeric keypad when **MouseKeys** is turned On.

key	purpose	comment
5	mouse button(s)	see other keys for mode change
8	move mouse up	holding down accelerates movement
2	move mouse down	“
4	move mouse left	“
6	move mouse right	“
1, 7, 9, 3	move mouse diagonally	“
- (dash)	make 5 key right button	stays in this mode until changed
/ (slash)	make 5 key left button	“
asterisk	make 5 key both left and right	“
0 (Ins)	make 5 key <i>drag</i> when clicked	“
. (Del)	make 5 key <i>drop</i>	returns 5 key to simple click functionality
Num Lock	Toggles NumKeys On or Off	