

# **Access Pack for Microsoft Windows**

Access via Keyboard, Mouse, and Communication or Computer Access Aids  
for Individuals with Disabilities

Developed by:

The Trace Research and Development Center  
Madison, WI.

Revision 2.0  
April 1992

For Microsoft Corporation

Mark Novak, Joe Schauer, and Gregg Vanderheiden

## **Table of Contents**

Introduction to Access Pack

Installation of Access Pack software

Automatic Startup of the Access Utility

Non-Windows Applications' Use of Access Features

The StickyKeys Feature

The Keyboard Response Features (SlowKeys, RepeatKeys and BounceKeys)

The MouseKeys Feature

The ToggleKeys Feature

The SerialKeys Feature

The ShowSounds Feature

The TimeOut Feature

## Introduction to Access Pack

Access Pack for Microsoft Windows makes it easier for users to control the keyboard and mouse when using Windows versions 3.0 or 3.1. It is designed primarily for individuals with motion-related disabilities who may have difficulty using the computer keyboard or mouse, and for individuals who prefer visual feedback in place of sounds.

Access Pack provides the following features:

|                   |   |
|-------------------|---|
| <b>StickyKeys</b> | allows single-fingered typing                                   |
| <b>SlowKeys</b>   | ignores keys which are pressed accidentally or for a short time |
| <b>RepeatKeys</b> | adjusts or disables the keyboard repeat rate                    |
| <b>BounceKeys</b> | ignores keys which are pressed too quickly or at the same time  |
| <b>MouseKeys</b>  | allows control of the mouse cursor using the keyboard           |
| <b>ToggleKeys</b> | provides audio queues when toggle keys are turned on or off     |
| <b>SerialKeys</b> | allows control of the computer using an alternate input device  |
| <b>ShowSounds</b> | provides a visual cue when sounds are generated                 |
| <b>TimeOut</b>    | turns off the Access features when they're not being used       |

These features can be used separately or in combination, and can be customized to suit the user's preference. They are all designed so that an individual who requires a feature can turn it on without assistance, as long as the Access Pack is installed. When all features are disabled, individuals who do not need them should not notice that the Access Pack is installed. Therefore, a computer on which Access Pack is installed can be used both by individuals who require it and those who do not.

It is recommended that Access Pack be installed on all computers which may be used by multiple people, such as public or workgroup machines.

The Access Pack software is fully compatible with most hardware configurations. Access Pack consists of three keyboard drivers and a mouse driver, which replace the standard Windows drivers which are used on most hardware. It also includes the Access Utility (ACCESS.EXE), a program that controls the Access function, and a few other files which the Access Utility uses.

## Installation of Access Pack software

To install Access Pack by hand

- 1) Exit Windows to return to the MS-DOS prompt.
- 2) Open the SYSTEM.INI file by using a text editor.

- 3) Find the section labeled [boot].
- 4) In the [boot] section, find the entry labeled "keyboard.drv=".
- 5) The following entries indicate that you are already running an older version of the Access Pack keyboard driver:

```
keyboard.drv=ap-kbd.drv
keyboard.drv=ap-kbdhp.drv
keyboard.drv=ap-kbdol.drv
```

If the entry in the [boot] section of your SYSTEM.INI file matches any of these, you need to copy the newer version of the corresponding .drv file to the Windows SYSTEM directory. For example, if the Access Pack software is located on a floppy disk in your A: drive, and your SYSTEM.INI file reads "keyboard.drv=ap-kbd.drv", and Windows is installed in the WINDOWS directory on drive C:, then you should type the following command at the MS-DOS prompt:

```
copy a:ap-kbd.drv c:\windows\system
```

The following entries indicate that you are running a keyboard driver compatible with a corresponding Access keyboard driver,

| <u>If your entry says:</u> | <u>change the entry to:</u> | <u>and copy file:</u> |
|----------------------------|-----------------------------|-----------------------|
| keyboard.drv=keyboard.drv  | keyboard.drv=ap-kbd.drv     | ap-kbd.drv            |
| keyboard.drv=kbdhp.drv     | keyboard.drv=ap-kbdhp.drv   | ap-kbdhp.drv          |
| keyboard.drv=kbdoli.drv    | keyboard.drv=ap-kbdol.drv   | ap-kbdol.drv          |

Instructions for copying the appropriate file to the Windows SYSTEM directory are explained in the preceding paragraph.

If your entry does not match any of those listed here, then your computer may not be compatible with the Access Pack software. Contact your computer manufacturer to find out if your computer can be used with any of the Access Pack keyboard drivers.

- 5) In the [boot] section, find the entry labeled "mouse.drv=".

If the entry says "mouse.drv=ap-mou.drv" then you are already running an older version of the Access Pack mouse driver. You need to copy the newer version of the AP-MOU.DRV file to your Windows SYSTEM directory.

If the entry says "mouse.drv=mouse.drv" then you are running standard keyboard drivers which can be replaced with the corresponding Access driver. Copy the file

AP-MOU.DRV to your Windows SYSTEM directory, and edit the entry in SYSTEM.INI to say "mouse.driv=ap-mou.driv".

If your entry does not match any of those listed here, contact your mouse manufacturer to find out if your mouse is compatible with the Microsoft mouse driver, or if the manufacturer has an Access driver available.

If you cannot install an Access mouse driver, you will not be able to use the Access Pack MouseKeys feature or control the mouse using the SerialKeys feature, but you will still be able to use the keyboard-related features.

- 6) Copy the following files from the Access Pack into your Windows directory:

```
ACCESS.EXE
ACCESS.WRI
ACCESS31.HLP
ACCESSU.DLL
IBMMKEYS.BMP
SKWIN.BMP
```

For example, if the Access Pack software is on a floppy disk in drive A: and Windows is installed in the WINDOWS directory on drive C:, then you should type the following commands at the MS-DOS prompt:

```
copy a:access.exe      c:\windows
copy a:access.wri     c:\windows
copy a:access31.hlp   c:\windows
copy a:accessu.dll    c:\windows
copy a:ibmmkeys.bmp   c:\windows
copy a:skwin.bmp      c:\windows
```

**Note:** If you are running Microsoft Windows version 3.0, you should copy the file ACCESS30.HLP instead of ACCESS31.HLP.

- 7) Start Windows, and then create a program item for the file ACCESS.EXE.

For more information about creating program items, see Chapter 3, "Program Manager," in the "Microsoft Windows User's Guide."

## Automatic Startup of the Access Utility

Once Windows has been configured to use the Access Pack keyboard driver, the access features can be activated whenever Windows is running. The individual features can be

enabled or disabled, or customized to fit the your preferences, by using the Access Utility (ACCESS.EXE).

The Access Utility can be started at any time to adjust these options. Instructions for starting applications are described in Chapter 2, "Application Basics", in the Windows 3.1 Windows User's Guide (or Chapter 2, ""Basic Skills" in the Windows 3.0 Windows User's Guide).

If you would like to have the Access Utility running at all times, you may have it started automatically every time you start Windows.

If you are running Windows 3.1, you should add a program item for the Access Utility to the "StartUp" program group in Program Manager. For detailed instructions, see Chapter 3, "Program Manager", in the Windows 3.1 Windows User's Guide.

If you are running Windows 3.0, you must edit a file called WIN.INI. To edit the WIN.INI file, you must have a word processor or text editor that has the ability to read, write and save a "text" or "ASCII" file. The WIN.INI file should be in the Windows directory (for example, c:\windows). The second or third line in the WIN.INI file contains the instruction "load = ". If a program name or some command appears after the "load = " instruction, just append the Access Utility onto the line by leaving a blank space and then type "access.exe" (without quotes). For example, to have the Access Utility and the Windows clock program start up as icons each time you start Windows, change the WIN.INI load line to read as follows:

```
load = clock.exe access.exe
```

If you would rather have the Access Utility start as an open window, append it to the next line of the WIN.INI file, which has the instruction "run = ".

## **Non-Windows Applications' Use of Access Features**

The Disability Access Features StickyKeys, RepeatKeys, SlowKeys, BounceKeys, MouseKeys, ShowSounds, ToggleKeys, and TimeOut will only function for Windows Applications. If you need similar access features in non-Windows applications, you will need to utilize other programs. Some of these other programs include Terminate and Stay Resident (TSR) programs such as AccessDOS. AccessDOS includes all of the above disability access features for DOS users and is available FREE from IBM (800) 426-7282.

## **The StickyKeys Feature**

StickyKeys is a feature that allows people who must type with 1 finger, a mouthstick or a

headstick to be able to use modifier keys (SHIFT, CTRL, and ALT keys). With StickyKeys turned on, a person can press a modifier key and then another key in sequence rather than at the same time, to get shifted (modified) characters. To turn StickyKeys on from the keyboard, tap either shift key five times (without disturbing the mouse). If you have the sound features of StickyKeys enabled, you will hear an up-siren tone. To turn off StickyKeys off from the keyboard, tap a shift key five times (you will hear a down-siren if the sound feature is enabled). StickyKeys can also be turned on or off from the StickyKeys dialog box under the "Adjust" menu.

StickyKeys operates in two modes: key latching mode and key locking mode. Tapping ONCE on a modifier key causes it to go into LATCHED mode. If you have the sound features of StickyKeys enabled, a short low beep - high beep will be heard. As soon as the next non-modifier key is pressed the modifier key(s) will be released. Tapping TWICE in succession on a modifier key will put it into LOCKED mode. If you have the sound features of StickyKeys enabled, you will hear a short low - high beep after the first tap and a single high beep after the second tap. Once a modifier key is LOCKED, it will stay "locked down" until that modifier key is hit a third time. Any and all of the modifier keys (SHIFT, CTRL, and ALT) can be latched or locked in combination.

For shared or public use computers, there is an optional "Turn off when pressing Two keys at once" feature to keep non-disabled users from being confused if StickyKeys is accidentally left on. Whenever an able-bodied typist uses a keyboard, they will hold the modifier key down and strike another key simultaneously. If the "Turn off when pressing Two keys at once" shutoff feature is enabled, StickyKeys will detect that two keys are down simultaneously and automatically turn the StickyKeys feature off.

Some people do not like to have keyboard sounds while others find them useful. The ability to turn the different feedback sounds on or off is provided in the StickyKeys dialog box under the "Adjust" menu. Also, it is possible to disable the LOCKED mode of StickyKeys described above if it is not desired.

On United States keyboards or when using "United States" as your country and keyboard layout, StickyKeys treats the modifier keys as identical pairs. In other words, StickyKeys will treat the left and right shift (SHIFT), the left and right control (CTRL), and the left and right alternate (ALT) keys the same. Treating the modifier keys as identical pairs allows the user greater flexibility. For example, if the user entered LOCKED mode by tapping the left shift key twice, they could perform their intended function and exit LOCKED mode by tapping the right shift key once, since the two modifier keys are treated the same.

On non United States keyboards or when not using "United States" as your country or keyboard layout, you may find that your particular international keyboard attempts to distinguish between the left and right modifier keys. In particular, your international keyboard may need to distinguish between the left and right alternate (ALT) keys. This is

typically done by giving the right alternate key a new name, instead of just "Alt", it is now called the "AltGr" key. "AltGr" is an abbreviation for "alt-graphics". For international keyboard layouts, StickyKeys will still allow the user to move freely between the left and right shift (SHIFT) or control (CTRL) keys, but you will notice a difference in the behavior of the left and right alternate (ALT) keys. The left alternate key will behave as the standard "Alt" key, typically used to quickly access a menu item, while the right alternate key will behave as the "AltGr" key, which many international keyboards use to access special characters important to their individual languages. Both the "Alt" and the "AltGr" keys can still be used in either LATCHED or LOCKED mode.

## The Keyboard Response Features

The Keyboard Response group is a set of functions that modify how the individual keys on the keyboard behave or respond when you press them. If the standard key-repeat feature of your keyboard is too fast, or you don't want it at all, RepeatKeys allows you to slow it down or turn it off. If you bump keys accidentally as you move around on the keyboard and would like to slow the keyboard response down so that it would only accept keys after they have been held down for awhile, SlowKeys will do this for you. If you have a tremor and accidentally type keys twice as you press or release them, BounceKeys can be used to make the computer ignore the extra tap on the keyboard.

The three functions under Keyboard Response, RepeatKeys, SlowKeys, and BounceKeys, are enabled and disabled together as a group. This is done to simplify turning the group on and off from the keyboard. Most people are only interested in using one or two of the functions in the group. This is not a problem. You can individually adjust the functions within this group such that when enabled, unwanted functions will have no effect.

There are two adjustments for the RepeatKeys function. The first setting, "Repeat Rate", allows you to adjust the speed at which the keys will repeat when you hold a key down. The smaller numbers (like 0.3) will make keys repeat faster. If you do not want the keys to repeat at all, then just set the Repeat Rate to OFF. If the Repeat Rate is set to OFF, the Delay until Repeat scroll bar will be disabled. To re-enable the Delay until Repeat, you must select a Repeat Rate other than OFF.

The second setting, "Delay until Repeat", allows you to set the length of time you must hold a key down before it will begin to repeat. If you have trouble releasing a key in time, then set this to a higher or longer setting.

(NOTE: For RepeatKeys to work within Windows 3.0, you must adjust the Control Panel keyboard setting for "Repeat Rate", a sliding bar, all the way to the left for "slow". For RepeatKeys to work within Windows 3.1, you must adjust the Control Panel keyboard setting for "Repeat Rate", a sliding bar, all the way to the left for "slow", and the



keyboard setting for "Delay Before First Repeat", another sliding bar, all the way to the right for "short".)

SlowKeys allows you to slow down the keyboard so that keys must be held down for a while before they will be accepted by the computer. This is useful for individuals who could use the keyboard but accidentally bump extra keys when they try to type. By slowing down the response of the keyboard, these extra key bumps are ignored and only the keys that are typed and held down are accepted. SlowKeys only has one setting called "Acceptance Delay". Acceptance Delay allows you to adjust the amount of time that you must hold a key down before it will be accepted by the computer. If you do not want the SlowKeys function to be active, then just set the Acceptance Delay to OFF.

Individuals who tremor or have impaired control of their hands may find that they type two or more of the same letter when they type. They bounce on the same key either when they are pressing it or when they release it, causing extra characters. BounceKeys allows the computer to ignore these quick bounces on the same key, so that only one key is accepted. There is only one setting for BounceKeys, "Debounce Time". The larger the time setting, the longer you will have to wait after you release a key before you can type the same key a second time (up to 2 seconds!). If you do not want the BounceKeys function, then just set the Debounce Time to OFF. When BounceKeys is on, if you want to type the same key twice you just need to pause a bit between the two key presses. BounceKeys does not prevent you from typing other keys quickly.

BounceKeys and SlowKeys cannot be used or both be active (turned on) at the same time. This may seem confusing, but if you have SlowKeys on (e.g. acceptance delay does not equal OFF), it will provide a BounceKeys like function on both the hit and release of each key. While SlowKeys would assist some individuals, it would hinder others since they may only tremor or bounce when they release a key. Therefore, these individuals would only want to have BounceKeys active.

Once you have set up the desired values under Keyboard Response (RepeatKeys, SlowKeys and BounceKeys), you need to enable group. If the Keyboard Response dialog box is open (e.g. under "Adjust" menu), then select ON at the top of the dialog box. This will turn on any of the Keyboard Response functions which you selected.

Sometimes you may want to enable or disable the Keyboard Response functions from the keyboard without having to open the dialog box. To do this just hold the right shift key down for 8 seconds. After about 4 seconds you will hear 3 short warning beeps. (These are provided just in case someone is accidentally resting their hand on the right shift key and doesn't really want to enable these functions - it's sort of a "get off the key" warning). Since you do want these functions enabled, just ignore these warning beeps and keep holding the key down. After about 4 seconds more you will hear an up-siren (e.g. provided Sound when turned on or off is enabled). You can release the right shift key now and the Keyboard Response functions will be enabled using the values you set for

RepeatKeys, SlowKeys and BounceKeys. You can disable the Keyboard Response group from the keyboard using this same procedure. When you disable the group you will hear a down-siren (e.g. provided Sound when turned on or off is enabled).

Some individuals are completely unable to operate the computer unless they can turn the keyboard repeat function off and/or turn SlowKeys on. These individuals could find themselves in a Catch-22, unable to use the computer to turn on the functions they need to operate the computer. There is a solution to this problem built into the Access Utility. There is an emergency enabling routine that allows you to turn the Keyboard Response functions on using only one key on the keyboard. To enable the emergency settings for Keyboard Response from the keyboard, you just hold down the right shift key (as you normally do to enable the Keyboard Response) except that you keep holding it down even after you hear the first up-siren. If you hold the right shift key down long enough, this is the sequence of events that will happen: after 4 seconds, 3 short warning beeps; after 4 more seconds, a single up-siren indicating that Keyboard Response has turned on with the settings you previously chose with the dialog box open or with the default settings; after 4 more seconds, you will hear 2 up-sirens which is the first level of emergency settings, RepeatKeys will have the Repeat Rate OFF (no key repeats), SlowKeys will be OFF, (no acceptance delay), and BounceKeys will have a Debounce Time of 1 second; after 4 more seconds (total of about 16 seconds), you will hear 3 up-sirens which is the second level of emergency settings, RepeatKeys will have the Repeat Rate OFF, SlowKeys will have the acceptance delay set to maximum or 2 seconds, and BounceKeys will be OFF. If an individual activates either of the two emergency settings, they should then open the Keyboard Response dialog box and change the settings to the values they need to operate the computer keyboard. After these changes have been made and saved, the user should exit the emergency settings (hold right shift key down for 8 seconds until the down-siren) and then re-activate Keyboard Response (hold right shift key down for 8 seconds until the up-siren). The Keyboard Response functions would then be turned on with the individuals newly saved settings.

Another very useful feature in the Keyboard Response functions is the setting called "Click when a key is Pressed". If this setting is checked and any of the Keyboard Response functions are active, you will hear a click when you press the key, or when the key repeats. For example, if SlowKeys is active, you would hear a key "click" sound when the key is pressed and also when the computer accepts the key. You may just wish to have a key "click" sound with every key you press while not using the Keyboard Response functions. To do this, set SlowKeys and BounceKeys to OFF, adjust RepeatKeys to your needs, and then turn on the Keyboard Response group. The "Click when a key is Pressed" feature provides additional user feedback. This can be very important when the keyboard has been modified to function or "respond" differently than the expected norm.

## **The MouseKeys Feature**

Some users do not have the physical control to operate a pointing device such as the mouse, yet need to access software which requires them to use the pointing device. MouseKeys provides a means to control the mouse cursor on the screen by using the numeric keypad on the keyboard. You turn MouseKeys on or off by simultaneously pressing the left ALT key, the left SHIFT key and the NUM LOCK key. When MouseKeys turns on, you will hear an up-siren if you have sounds enabled. (If you are using only one finger, a mouthstick, or a headpointer to operate the computer, the easiest way to activate MouseKeys is to first activate StickyKeys by tapping either SHIFT key 5 times. You can then hit the three keys in sequence rather than simultaneously.) Once MouseKeys is turned on, the numeric keypad becomes a mouse control pad. Each key on the keypad performs a different mouse function. The "5, +, 0, and ." keys operate the buttons on the mouse. Use the "5" key when you want to click a mouse button once. To double-click a mouse button, use the "+" key. When you want to hold a mouse button down while you use the mouse (as for dragging), press the "0" key to lock the mouse button down and the "." key to release the mouse button. On 101-key keyboards, the "/" makes the left mouse button the currently active button, the "-" makes the right mouse button the currently active button, and the "\*" makes both mouse buttons the currently active button. On older keyboard with fewer keys, it may not be possible to make both mouse buttons active. The keys "1, 2, 3, 4, 6, 7, 8, and 9" are used to move the mouse cursor around the screen in the directions suggested by their location on the numeric keypad. For example, pressing the "1" key will move the mouse cursor down and to the left while pressing the "8" key will move the mouse cursor straight upward on the screen. Holding a direction key down will cause the mouse cursor to continue to move in the respective direction, slowly at first, and then increasing to the maximum speed. The mouse pointer will then continue at maximum speed until the key is released. Both the maximum speed of the mouse cursor and the time it takes to get to maximum speed are adjustable in the MouseKeys dialog box under the "Adjust" menu.

(NOTE: If SlowKeys is active, all the MouseKey control keys will respond according to the amount of "Acceptance Delay" selected.)

While MouseKeys is on, you can use the NUM LOCK key to toggle the MouseKeys control pad back to a numeric keypad and vice versa. This feature is especially useful if you are using a laptop or notebook computer which doesn't have a separate numeric keypad. On these computer keyboards, the numeric keypad is usually overlaid on top of the standard "qwerty" keyboard. Tapping once on the NUM LOCK key to toggle between either a numeric keypad or a mouse control pad makes it very convenient for the user. When toggling between a mouse control pad and the numeric keypad, the numeric keypad will be in the state it was in before MouseKeys was started. For example, if you were using the numeric keypad for number entry before starting MouseKeys, when you toggle out of MouseKeys with the NUM LOCK key, you will be able to enter numbers with the numeric keypad. If you were using the numeric keypad as a cursor keypad before starting MouseKeys, when you toggle out of MouseKeys using the NUM LOCK

key, you will have a cursor keypad. MouseKeys will work in Windows 3.0 and Windows 3.1 in conjunction with the standard Microsoft or IBM PS/2 mouse. You can use the standard mouse to move quickly about the screen and then use MouseKeys to move more precisely (unit by unit) to your final destination. Some people cannot use the standard mouse while simultaneously holding down the mouse button. You can use MouseKeys to lock down the currently active mouse button, move the mouse cursor using MouseKeys or the real mouse, and then release the mouse button using MouseKeys.

Note: This version of the Access Utility (3.0.a) does not require the user to have a Microsoft or PS/2 mouse connected to the computer in order for MouseKeys to work while running Windows.

## **The ToggleKeys Feature**

People with visual impairments may not be able to see the lights on the keyboard that indicate CAPS LOCK, NUM LOCK and SCROLL LOCK status. ToggleKeys provides a solution to this by emitting a tone whenever these keys are pressed. If ToggleKeys is enabled, when you press one of these keys and it turns on, you will hear a high beep. When you press one of these keys and it turns off, you will hear a low beep. ToggleKeys is turned on from the keyboard by pressing the NUM LOCK key and holding it down for about 5 seconds. When ToggleKeys turns on, you will hear an up-siren if sound is enabled. You turn ToggleKeys off from the keyboard the same way, just hold the NUM LOCK key for about 5 seconds and you will hear a down-siren. ToggleKeys can also be turned on or off from the ToggleKeys dialog box under the "Adjust" menu. ToggleKeys will also function with some keyboards which do not have the status indicator lights for the CAPS LOCK, NUM LOCK and SCROLL LOCK keys. The audible low and high beeps can be very useful for all operators when using this style of keyboard.

## **The SerialKeys Feature**

Some people cannot use the keyboard and mouse at all due to a physical impairment, and must use some specially designed input device to perform the functions of the keyboard and the mouse. In some cases these special input devices can be connected to a computer's serial port (the connector commonly used for attaching a modem or printer). Simply connecting the special input device to the computer's serial port, however, is not enough to allow you to emulate or control keyboard and mouse functions. The computer is designed to receive keyboard and mouse signals through its keyboard and mouse ports, not through its serial port.

SerialKeys allows you to control keyboard and mouse functions through the serial port. This is done by sending the right series of letters and commands from the special input device to the computer's serial port. The regular keyboard and mouse can still be used while SerialKeys is being used. However, typing at the standard keyboard while simultaneously entering commands using SerialKeys may cause SerialKeys to loose track

of the status of various keys on the standard keyboard.

The most commonly used special input device for SerialKeys is a communication aid: an electronic device often used by people who can't communicate by speech alone because of a physical disability that affects their speech. Using SerialKeys requires you to program specific key definitions into your special input device. You will need the manual for your device in order to carry out this programming.

The instructions below tell you how to set up SerialKeys with a device such as a communication aid (we just refer to the device as "the aid" in these instructions). These instructions tell you how to hook up your aid to the computer, how to set up SerialKeys on the computer, and what letters or commands to send so you can type any key on the keyboard and make any mouse movement you want from the aid. Follow the instructions step-by-step, and you should be successful in hooking up your aid and using SerialKeys.

NOTE: These instructions do NOT tell you how to program your own aid. You will need to consult the manual for your aid, or contact the manufacturer if you have questions.

The steps in these instructions are numbered from 1 to 20; steps 1 through 10 discuss basic skills, and steps 11 through 20 discuss advanced topics. If you are the user of the aid, someone may need to assist you the first time you set up SerialKeys.

### **Step 1: Determine if your communication aid will work with SerialKeys**

You need to make sure your aid will work with SerialKeys. You should be able to answer "yes" to all six questions:

- 1) Can you program selections on your aid to have unique words and sentences stored under them? Most electronic communication aids these days will let you program any series of characters (letter, numbers, symbols) you want for a particular selection.
- 2) Can your aid store at least 84 programmable selections? You will need at least one selection space for each key on the computer's keyboard, plus more if you want to use the mouse. There are 84 keys on the standard IBM AT keyboard, and now many keyboards have 101 keys. You don't need to be able to store all of the keys and mouse actions on the same level (or overlay) on your aid. You may, for instance, want to use one level for the keyboard and one for the mouse.
- 3) Does your aid have a serial port? This is the connection sometimes used to hook up a printer. It is sometimes called an RS-232 serial port.
- 4) Can you program selections on the aid so that when they are selected, they will automatically be sent to the serial port? Many aids have this capability, since it can be useful for printing messages. If you have a printer connected to your serial port, typing

h-e-l-l-o on your aid and sending it out the serial port will make "hello" print out on the printer. If you have your aid's serial port connected to the computer's serial port, and you have SerialKeys working, typing h-e-l-l-o on your aid and sending it out the aid's serial port will cause the word "hello" to appear on the computer--just as if it had been typed on the computer's keyboard.

5) Does your computer have a serial port? This is sometimes called a "COM" port in computer jargon. If you don't have one, you can probably add one to your computer. If you have a serial port but you are using it for something else, such as connecting a printer or a modem, you will have to:

- disconnect the printer or modem, OR
- equip your computer with another serial port.

6) Can your communication aid send information through its serial port at the rate of 300 baud? (The term 300 baud refers to the speed at which characters are transmitted.) Most aids can send information at 300 baud, but check and make sure. Your aid may be able to send information faster than 300 baud as well, and that's OK. SerialKeys can "talk to" an aid that's sending information at 300, 600, 1200, 2400, 4800, or 9600 baud. However, 300 baud is the recommended speed setting. If your aid requires a higher baud rate, you will need to be aware of some additional operating features.

### **Step 2: Connect your aid to your computer**

Just plug one end of the serial cable into the serial port on your aid and the other end into the serial port on your computer. If you don't already have a serial cable, you will need to obtain it from the manufacturer of your aid.

NOTE: Your aid may require an adapter to match the connector on your computer. In addition, your aid may require a NULL modem adapter to talk properly with your computer.

### **Step 3: Configure your aid for serial transmission**

Your aid and SerialKeys must be talking to each other at the same speed (baud rate) for them to communicate at all. We recommend choosing a speed of 300 baud. You must also make sure your aid is configured correctly. It must be communicating at 300 baud, with the following format:

- 1 start bit
- 8 data bits
- 1 stop bit
- no parity.

You don't have to know what this means, you just have to make sure your aid is

configured that way. Check the manual of your aid to find out how to do it. In addition, most aids have a special selection or a switch for causing the characters stored in your aid to be transmitted out the serial port. Make sure you "turn on" this function. Other aids may require you program special codes to set up the serial port. Again, please read your manual to find out how to do this.

#### **Step 4: Activate SerialKeys**

Before you can use SerialKeys, you need to make sure SerialKeys is active within Windows on your computer. Open the Access Utility window, and under the "Adjust" menu item, select SerialKeys. The SerialKeys dialog box will then open. Then select the Serial Port, either Com port 1, Com port 2, Com port 3, or Com port 4, and select the Baud Rate (suggest 300 baud to start). Then turn SerialKeys ON and exit the dialog box. You may also want to save this setting change under the "File" menu.

NOTE: Only version 3.0.a of the Access Utility provides the user with the option of choosing between Com ports 1 through 4. The Access Utility does not attempt to verify whether or not your computer hardware actually supports Com ports 1 through 4, but rather depends on Windows to make such a determination. Before attempting to connect your communication aid, you may want to review your computer manuals to be sure it supports your choice of either Com port 1, 2, 3, or 4.

#### ***Learn to program keys and mouse actions on your aid***

The next series of steps will tell you how to program each different type of key or mouse movement, and give you examples which you can try programming. This way, you can check and make sure that your aid and computer are set up properly, and that you understand how to program the aid, before you program the complete set of keys and mouse movements. There are four basic types of SerialKeys actions you can program into any selection (position) on your aid:

- 1) basic keys,
- 2) special keys,
- 3) modifier keys,
- 4) mouse movements.

Before you start typing keys, send three null characters from your aid to the computer. This resets SerialKeys. (The null character is different from a zero; usually it can be made on your aid by typing control-@.)

#### **Step 5: Typing basic keys**

Each key on your computer keyboard is given a keyname, and it is this keyname that you must program into your aid if you want to be able to type that key. Many of the

keynames consist of a single character. These we are calling "basic keys."

```
`1 2 3 4 5 6 7 8 9 0 -
q w e r t y u i o p [ ]
a s d f g h j k l ; '
z x c v b n m , . / \
```

To type one of these keys, you just send that single character out the serial port of the aid. For example, to type the word "hello", you simply select the "h" keyname, the "e" keyname, the "l" keyname, another "l" keyname, and finally an "o" keyname. Most aids have all of these single character keynames programmed into them so there is no additional programming for you to do for these keys.

### Step 6: Typing special keys

If you look at the computer keyboard, you will find that there are many keys that do not have single character keynames. For example, the Num Lock and the Enter key each have more than a single character name. We call these keys special keys since you have to do something different if you want to type them using your communication aid. To type special keys, you must program a sequence of characters into a single selection on your communication aid. The sequence for special keys is different for each key, but always follows two rules:

- 1) The sequence must begin with the Escape character.
- 2) The sequence must end with a period.

The Escape character usually can be generated on your aid by selecting the Control function along with the left bracket ([). On some aids, this will appear as ^[. Check the instruction manual of your aid if you're unsure of how to make the Escape character on it. We will refer to the Escape character by putting "esc" between two angle brackets, like this: <esc>. Remember: when you see <esc> in a sequence of characters, it just means ONE character (the Escape character), not five characters, as it might appear.

To program the name of a special key into your aid, program the Escape character first, then the correct name for the special key, then a period. These should all be programmed as one word or selection on your aid. Then when you need the special keys, the entire string would be sent out to the serial port of the computer by only making one selection on your aid.

For example, to type the Enter key, the character sequence would be:

```
<esc>enter.
```

The escape character comes first, followed by the keyname for the key that you want to



press, followed by a period (.). SerialKeys has official names for each of the special keys on your keyboard. Since the sequences for special keys are somewhat long, you will want to program an entire sequence for a special key into a single selection on your aid. Then you can label this selection (e.g. Enter) and have it ready for when you need to type the key. In fact, you will probably want to program the sequences for all the special keys on your computer keyboard into selections on your aid. The time spent doing this will be made up many times once you begin using your aid to access the computer. See Steps 9 and 10 to learn how to set up the complete keyboard, and for a table of all the keynames.

### **Step 7: Typing Modifier Keys (Shift, Control, Alt)**

There are three special keys which have to be typed differently from the other special keys: Shift, Control (Ctrl), and Alternate (Alt). These three keys are called the modifier keys, since they don't do anything themselves but modify the action of another key. Shift, for instance, will make a small "a" into a capital "A.". On an ordinary keyboard, the modifier key is held down while the other key is pressed. SerialKeys imitates this action by using a command called "hold". Here's the sequence you have to type on your aid in order to type a modifier key:

- 1) The Escape character
- 2) Comma (,)
- 3) The word "hold"
- 4) Another comma (,)
- 5) The name for the modifier key (see Step 10, below)
- 6) Period (.)
- 7) The key you want to modify.

The following command string would be used to send a capital A on your computer using SerialKeys.

```
<esc>,hold,shift.    (holds the shift down)
a                    (while you send the "a" key)
```

Remember, <esc> means the Escape character on your aid. And don't forget the commas and the period. If you've typed the sequence successfully, a capital "A" should appear on your computer, just as if you had held down the Shift key on the regular keyboard and then pressed the "a" key.

Most keyboards have two Shift keys, a right shift key and a left shift key. You can use different keynames (lshift and rshift) for the two keys to distinguish between them if you wish. Some keyboards also have right and left Control keys and right and left Alternate keys. SerialKeys also distinguishes between these. See Steps 9 and 10 to learn how to set up the complete keyboard, and for a table of all the keynames.

## Step 8: Moving and clicking the mouse

If your computer is equipped to use a mouse (IBM PS/2 or Microsoft mouse), you can operate all the mouse functions from your aid as well, using SerialKeys. If you're not interested in performing mouse functions, skip over to Step 9 and 10.

Before you do any mouse functions through SerialKeys, you should send the "moureset" (mouse reset) command. This should be done whenever you start up an application program (such as a word processing program) that uses the mouse. Just send this command from your aid.

```
<ecs>,moureset.
```

The mouse pointer should move to the upper left hand corner of the screen.

### *Moving the mouse*

You can do the equivalent of moving the mouse by sending a "move" command from your aid to the computer. Here are the parts of a move command, in the proper order:

- 1) the Escape character (see Step 4 if you don't know what this is)
- 2) a comma
- 3) the word "move"
- 4) another comma
- 5) a plus or minus sign followed by a number  
(for horizontal movement)
- 6) another comma
- 7) another plus or minus sign followed by a number  
(for vertical movement)
- 8 a period.

The move command takes two numbers after it: first the motion in the horizontal direction and then the motion in the vertical direction. There must be either a plus (+) or a minus (-) sign before each number, unless the number is a zero. Positive numbers move the mouse pointer to the right or down. Negative numbers move the mouse pointer left or up.

Examples:

```
<esc>,move,+10,-20. moves 10 units to right and  
20 units up
```

```
<esc>,move,-10,+20. moves 10 units to left and  
20 units down
```

### ***Clicking the Mouse***

In order to click the button on a mouse, you use the "click" command. The commands should be typed as you see below. Remember, <esc> means a single "Escape" character.

Examples:

<esc>,click,left.      clicks the left button  
<esc>,click,right.     clicks the right button

### ***Double-clicking the mouse***

You can double-click the mouse using the same commands shown above, but with the command "dblclick" in place of the command "click":

Example:

<esc>,dblclick,left.

### ***Clicking and Dragging with the Mouse***

SerialKeys lets you do the equivalent of holding the mouse button down while moving the mouse. This is done using the "moulock" (mouse lock) and "mourel" (mouse release) commands, together with the "move" command. If, for example, you want to select text in a word processing program that uses the mouse, you would:

- 1) Move the mouse to the beginning of the text you want to select, using the "move" command as described above.
- 2) To press the mouse button down and keep it down, give the "mouse lock" command:

<esc>,moulock,left.

- 3) Move the mouse to the end of the text you want to select, using the "move" command as described above.
- 4) To release either button (or both), give the "mouse release" command:

<esc>,mourel.

## **Step 9: Plan the layout for your aid**

Steps 5-8 above showed you what sequences of characters to send in order to type basic keys, special keys and modifier keys, and to do mouse actions. Now you will want to

program a selection on your aid for each key or mouse action you want to use. But before you program all of the keys and mouse actions into your aid, you will probably want to decide how to lay out the selections on your aid. Look at the number of different keys you may want to have (see the table under Step 10). You will probably want mouse movements in addition? If your aid stores vocabulary in levels, decide if you want keyboard and mouse actions on separate "levels" in your aid. If your aid uses combinations of symbols to store vocabulary, what combinations will you want to use for keyboard and mouse functions?

### Step 10: Programming your aid for keyboard and mouse functions

The following table lists character sequences you send from your aid to simulate the individual keyboard keys. In some cases there are several character sequences which all send simulate the same key.

In this table, the term "p/r" is used to indicate a key being pressed and immediately released. "Toggled" indicates that a key such as CAPS LOCK will be turned on if currently off, or turned off if currently on. "Pressed down" indicates that the key will be pressed and remain held down during the next key simulated.

| <u>Key name</u> | <u>Send from aid</u> | <u>Action(s)</u>                      |
|-----------------|----------------------|---------------------------------------|
| alt             | <esc>,hold,alt.      | alternate key pressed down            |
| backspace       | <esc>backspace.      | backspace key p/r                     |
| bksp            | <esc>bksp.           | backspace key p/r                     |
| capslk          | <esc>capslk.         | caps lock key toggled                 |
| capslock        | <esc>capslock.       | caps lock key toggled                 |
| comma           | <esc>comma.          | comma key p/r                         |
| control         | <esc>,hold,control.  | control key pressed down              |
| ctrl            | <esc>,hold,ctrl.     | control key pressed down              |
| del             | <esc>del.            | numeric pad del key p/r, "." or "del" |
| delete          | <esc>delete.         | delete key p/r                        |
| divide          | <esc>divide.         | divide key p/r, "/"                   |
| down            | <esc>down.           | down arrow key p/r                    |
| end             | <esc>end.            | end key p/r                           |
| enter           | <esc>enter.          | enter or return key p/r               |
| esc             | <esc>esc.            | escape key p/r                        |
| escape          | <esc>escape.         | escape key p/r                        |
| f1              | <esc>f1.             | function 1 key p/r                    |
| f2              | <esc>f2.             | function 2 key p/r                    |
| f3              | <esc>f3.             | function 3 key p/r                    |
| f4              | <esc>f4.             | function 4 key p/r                    |
| f5              | <esc>f5.             | function 5 key p/r                    |
| f6              | <esc>f6.             | function 6 key p/r                    |
| f7              | <esc>f7.             | function 7 key p/r                    |

|            |                  |   |
|------------|------------------|---|
| f8         | <esc>f8.         | function 8 key p/r                          |
| f9         | <esc>f9.         | function 9 key p/r                          |
| f10        | <esc>f10.        | function 10 key p/r                         |
| f11        | <esc>f11.        | function 11 key p/r                         |
| f12        | <esc>f12.        | function 12 key p/r                         |
| home       | <esc>home        | home key p/r                                |
| ins        | <esc>ins.        | numeric pad ins key p/r, "0" or "ins"       |
| insert     | <esc>insert.     | insert key toggled                          |
| kp*        | <esc>kp*.        | numeric pad star key p/r, "*"               |
| kp+        | <esc>kp+.        | numeric pad plus key p/r, "+"               |
| kp-        | <esc>kp-.        | numeric pad minus key p/r, "-"              |
| kp/        | <esc>kp/.        | numeric divide key p/r, "/"                 |
| kp0        | <esc>kp0.        | numeric pad 0 key p/r, "0" or "ins"         |
| kp1        | <esc>kp1.        | numeric pad 1 key p/r, "1" or "end"         |
| kp2        | <esc>kp2.        | numeric pad 2 key p/r, "2" or "down arrow"  |
| kp3        | <esc>kp3.        | numeric pad 3 key p/r, "3" or "pgdn"        |
| kp4        | <esc>kp4.        | numeric pad 4 key p/r, "4" or "left arrow"  |
| kp5        | <esc>kp5.        | numeric pad 5 key p/r, "5"                  |
| kp6        | <esc>kp6.        | numeric pad 6 key p/r, "6" or "right arrow" |
| kp7        | <esc>kp7.        | numeric pad 7 key p/r, "7" or "home"        |
| kp8        | <esc>kp8.        | numeric pad 8 key p/r, "8" or "up arrow"    |
| kp9        | <esc>kp9.        | numeric pad 9 key p/r, "9" or "pgup"        |
| kpdel      | <esc>kpdel.      | numeric pad . key p/r, "." or "del"         |
| kpdelete   | <esc>kpdelete.   | numeric pad . key p/r, "." or "delete"      |
| kpdivide   | <esc>kpdivide.   | numeric pad divide key p/r, "/"             |
| kpdown     | <esc>kpdown.     | numeric pad 2 key p/r, "2" or "down arrow"  |
| kpdp       | <esc>kpdp.       | numeric pad . key p/r, "." or "del"         |
| kpend      | <esc>kpend.      | numeric pad 1 key p/r, "1" or "end"         |
| kpenter    | <esc>kpenter.    | numeric pad enter key p/r                   |
| kphome     | <esc>kphome.     | numeric pad 7 key p/r, "7" or "home"        |
| kpins      | <esc>kpins.      | numeric pad 0 key p/r, "0" or "ins"         |
| kpinsert   | <esc>kpinsert.   | numeric pad 0 key p/r, "0" or "ins"         |
| kpleft     | <esc>kpleft.     | numeric pad 4 key p/r, "4" or "left arrow"  |
| kpmidl     | <esc>kpmidl.     | numeric pad 5 key p/r, "5"                  |
| kpminus    | <esc>kpminus.    | numeric pad minus key p/r, "-"              |
| kppagedown | <esc>kppagedown. | numeric pad 3 key p/r, "3" or "pgdn"        |
| kppageup   | <esc>kppageup.   | numeric pad 9 key p/r, "9" or "pgup"        |
| kppgdn     | <esc>kppgdn.     | numeric pad 3 key p/r, "3" or "pgdn"        |
| kppgup     | <esc>kppgup.     | numeric pad 9 key p/r, "9" or "pgup"        |
| kpplus     | <esc>kpplus.     | numeric pad plus key p/r, "+"               |
| kpriight   | <esc>kpriight.   | numeric pad 6 key p/r, "6" or "right arrow" |
| kpslash    | <esc>kpslash.    | numeric pad divide key p/r, "/"             |
| kpstar     | <esc>kpstar.     | numeric pad star key p/r, "*"               |
| kptimes    | <esc>kptimes.    | numeric pad star key p/r, "*"               |

|              |                          |  |
|--------------|--------------------------|--|
| kpup         | <esc>kpup.               | numeric pad 8 key p/r, "8" or "up arrow" |
| lalt         | <esc>,hold,lalt.         | left alternate key pressed down          |
| lcontrol     | <esc>,hold,lcontrol.     | left control key pressed down            |
| lctrl        | <esc>,hold,lctrl.        | left control key pressed down            |
| left         | <esc>left.               | left arrow key p/r                       |
| leftalt      | <esc>,hold,leftalt.      | left alternate key pressed down          |
| leftcontrol  | <esc>,hold,leftcontrol.  | left control key pressed down            |
| leftctrl     | <esc>,hold,leftctrl.     | left control key pressed down            |
| leftshift    | <esc>,hold,leftshift.    | left shift key pressed down              |
| lshift       | <esc>,hold,lshift.       | left shift key pressed down              |
| multiply     | <esc>multiply.           | star key p/r, "*"                        |
| numlk        | <esc>numlk.              | num lock key toggled                     |
| numlock      | <esc>numlock.            | num lock key toggled                     |
| pagedown     | <esc>pagedown.           | page down key p/r                        |
| pageup       | <esc>pageup.             | page up key p/r                          |
| period       | <esc>period.             | period key p/r, "."                      |
| pgdn         | <esc>pgdn.               | page down key p/r                        |
| pgup         | <esc>pgup.               | page up key p/r                          |
| ralt         | <esc>,hold,ralt.         | right alternate key pressed down         |
| rcontrol     | <esc>,hold,rcontrol.     | right control key pressed down           |
| rctrl        | <esc>,hold,rctrl.        | right control key pressed down           |
| ret          | <esc>ret.                | return or enter key p/r                  |
| return       | <esc>return.             | return or enter key p/r                  |
| right        | <esc>right.              | right arrow key p/r                      |
| rightalt     | <esc>,hold,rightalt.     | right alternate key pressed down         |
| rightcontrol | <esc>,hold,rightcontrol. | right control key pressed down           |
| rightctrl    | <esc>,hold,rightctrl.    | right control key pressed down           |
| rightshift   | <esc>,hold,rightshift.   | right shift key pressed down             |
| rshift       | <esc>,hold,rshift.       | right shift key pressed down             |
| scroll       | <esc>scroll.             | scroll lock key toggled                  |
| scrolllock   | <esc>scrolllock.         | scroll lock key toggled                  |
| shift        | <esc>,hold,shift.        | shift key pressed down                   |
| space        | <esc>space.              | space key p/r                            |
| tab          | <esc>tab.                | tab key p/r                              |
| tilde        | <esc>tilde.              | tilde key p/r, "~"                       |
| up           | <esc>up.                 | up arrow key p/r                         |

### Step 11: Troubleshooting

If you are using SerialKeys and your aid stops sending keys successfully, try:

- 1) Checking to make sure you included any necessary periods in your keynames.
- 2) Sending three null characters. (The null character is different from a zero; usually it

can be made on your aid by typing control-@.

3) Resetting both the aid and SerialKeys to 300 baud. (If there is a communication difficulty, SerialKeys may automatically reset itself to 300 baud, making it unable to communicate with your aid if your aid is sending at a different rate.)

### **Step 12: Advanced Topics**

We recommend programming your aid to use the IBM Enhanced Keyboard (101 keys). Even if your computer does not have this keyboard, SerialKeys will function as the 101-key keyboard. This may be to your advantage, as some software packages will recognize the additional keys, enabling you to access additional features. If you choose to program the 83 or 84 key keyboard, there are a few exceptions you should be aware of:

1) To type the Break function, you would ordinarily hold the Control key and press the Scroll Lock key. For SerialKeys, hold the Control key and press the Pause key.

2) To type the Pause function you would ordinarily hold the Control key and press the Num Lock key. For SerialKeys, just press the Pause key.

### **Step 13: Advanced Mouse Movements**

It's a good idea to at least program some square or selection on your communication aid to move the mouse in the four directions by 1, by 10, and by 100 units. This will allow you to make small, fine movements and large, fast movements.

Examples:

|                    |                                    |
|--------------------|------------------------------------|
| <esc>,move,+1,0.   | moves mouse cursor 1 unit right    |
| <esc>,move,-1,0.   | moves mouse cursor 1 unit left     |
| <esc>,move,0,+1.   | moves mouse cursor 1 unit down     |
| <esc>,move,0,-1.   | moves mouse cursor 1 unit up       |
| <esc>,move,+10,0.  | moves mouse cursor 10 units right  |
| <esc>,move,-10,0.  | moves mouse cursor 10 units left   |
| <esc>,move,0,+10.  | moves mouse cursor 10 units down   |
| <esc>,move,0,-10.  | moves mouse cursor 10 units up     |
| <esc>,move,+100,0. | moves mouse cursor 100 units right |
| <esc>,move,-100,0. | moves mouse cursor 100 units left  |
| <esc>,move,0,+100. | moves mouse cursor 100 units down  |
| <esc>,move,0,-100. | moves mouse cursor 100 units up    |

#### **Step 14: Advanced Notes on Resetting SerialKeys**

If you are using SerialKeys in a multi-user environment, the first command you send to SerialKeys should be the reset command. This will insure that SerialKeys will be at 300 baud and ready to accept your keyboard and mouse actions. The reset command consists of sending 3 Null (ASCII 0) characters with your aid configured to 300 baud. A Null character can usually be generated on your aid by selecting the control @.

#### **Step 15: Advanced Notes on using Lock and Release**

This command can be used to hold a key down and lift it back up again as separate actions. Its primary usefulness is in using SerialKeys in combination with MouseKeys.

- 1) Turn on MouseKeys (see Help on MouseKeys).
- 2) Decide which direction you want to move the mouse. Find out what numeric keypad key moves the mouse in that direction in MouseKeys.
- 3) Send a lock command with that keyname. For example to move the mouse cursor to the right:

<esc>,lock,kpright. or <esc>,lock,kp6.

- 4) When the mouse pointer has moved as far as you want in that direction, send the release command:

<esc>,rel.

You can program the "lock" part and the "release" part of this sequence each under a different selection on your aid, so you don't have to type them out in full each time you use them. Make sure to include the commas and periods.



Note: MouseKeys also allows you to hold and release a mouse button or click a mouse button. See the "Help" menu section on MouseKeys if you want to use these features with SerialKeys.

### **Step 16: Advanced Notes on using the Keyboard Combine Command**

If you want to program a modifier key and some other key under a single selection on your aid, use the combine command to put several keystrokes under one selection on your aid. This can be useful for common multiple-key command combinations required by software. There must be commas between the keys and a period at the end. No more than five keys can be combined.

Example: <esc>,combine,shift,ctrl,enter.

### **Step 17: Advanced Notes on using the Mouse Goto Command**

Moves the mouse to a specified location. You should send the "moureset" command first. Both the horizontal and vertical direction numbers require only a + sign. (See moving the mouse, Step 8 above.)

Example:

<esc>,goto,+20,+25.

### **Step 18: Advanced Notes on using the Mouse Anchor Command**

The mouse anchor command can be used to anchor the mouse cursor to a position within a Windows Application. An example of using the mouse anchor command to mark a current window position, go to a new location and click the mouse to select a new tool, go to another new location and click the mouse to select another color, and then return to the exact location you left to use that tool, is shown below. If you change the Active Window between setting the mouse anchor and returning to that anchor, the mouse anchor command will not work.

Example:

|                     |  |
|---------------------|--|
| <esc>,mouanchor.    | set the position or anchor the cursor                          |
| <esc>,goto,+10,+10. | go to a new location   |
| <esc>,click.        | click selects a new tool in this application at location 10,10 |
| <esc>,goto,+50,+10. | go to a new location   |
| <esc>,click.        | click selects a new color in this application at location      |
| 50,10               |  |
| <esc>,mouanchor.    | return to where you left from, anchor released                 |

### **Step 19: Advanced Notes on using the Baudrate Command**

The baudrate command allows you to change the baud rate of SerialKeys from your aid. The possible baud rates are 300, 600, 1200, 2400, 4800, and 9600. This command is never absolutely necessary, since you can also set the baud rate of SerialKeys by going into the "Adjust" menu.

Example:

```
<esc>,baudrate,300.
```

#### ***Using higher baud rates***

If you decide to run SerialKeys at a baud rate faster than 300 baud, you must be aware of the special automatic reset feature of SerialKeys. Whenever SerialKeys receives three consecutive characters with a transmission error, it will automatically reset to 300 baud. This is to enable users in a multi-user environment to put SerialKeys in a known state (300 baud). SerialKeys will signal any transmission problem with a short beep. If SerialKeys resets to 300 baud, it will make a long beep. SerialKeys will also signal a baudrate change with a long beep.

### **Step 20: Additional Technical notes**

- 1) SerialKeys uses hardware handshaking (DTR/RTS) and software (XON/OFF) handshaking to control the flow of characters from the aid. Characters may be lost if the aid ignores the handshaking signals.
- 2) SerialKeys allows Windows to manage the Com ports including the hardware interrupt lines. Windows may or may not allow another device to use or share the interrupt line that SerialKeys is using. This means that if for example, you have SerialKeys turned on using Com port 1, you probably can not use the Windows Terminal program on Com port 1. Also, if you have SerialKeys on Com 1, and another application is experiencing problems while using Com 3, you may need to move the application or SerialKeys to Com 2 or Com 4. See your computer manual for details on how Com 1/3 and Com 2/4 should be set, if your computer supports additional Com ports 3 and 4.

### **The ShowSounds Feature**

When the Access Utility is used by individuals with a hearing impairment, or by individuals operating the computer in a noisy environment, ShowSounds can be used to alert the computer operator when the computer operating system or an application has produced a sound. ShowSounds can alert the operator by either flashing the entire Window Desktop once (e.g. your main screen), or by flashing only the current Active

Window Caption Bar. ShowSounds can not be set to do both at the same time. If there are no current Active Windows, ShowSounds cannot flash any Caption Bars. Once Caption Bar flashing has started, you can tell ShowSounds to stop by typing any key on the keyboard. ShowSounds can only be enabled from the ShowSounds dialog box under the "Adjust" menu.

(NOTE: ShowSounds may not detect sounds produced by the computer if they are very short in duration or length.)

(NOTE: The Window Desktop option is only supported on CGA, EGA and VGA displays. On other displays the option will be grayed out and unavailable.)

## **The Timeout Feature**

When the Access Utility is used on a shared computer, it is sometimes useful to have an automatic timeout feature that would turn the access functions off if the computer were left idle for awhile. The TimeOut function provides this capability. There are two settings for the TimeOut function, the "Allow ACCESS options to Turn Off if unused for" setting and whether or not the TimeOut function should make an audible sound when it turns all the keyboard, mouse, and sound functions off. The TimeOut function can only be turned on or off from the TimeOut dialog box under the "Adjust" menu.

(NOTE: The TimeOut feature does not turn off the SerialKeys function.)