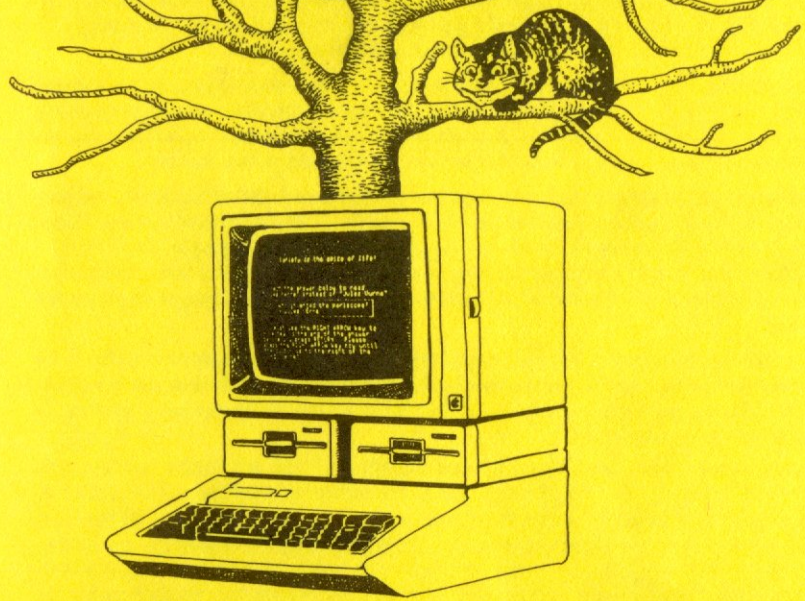
STORYWORKS



By
Robert C. Moore

TI&IE
P.O. Box 6229
Lincoln, NE 68506

Welcome to StoryWorks

Imagine being able to use the AppleWorks word processor to create on-screen quizzes and tutorials complete with sound effects. Think of how convenient it would be if you could have your students use these AppleWorks files and let the computer score their work and tell them (or you) how they did. Wouldn't it be fun to have your students create and read on-screen "twist-a-plot" adventures, adventures that feature students from your class in the main roles? Now you can do this and much more with StoryWorks.

StoryWorks permits you to use the AppleWorks word processor to create incredibly powerful "hypertext" applications (stacks). This means you can use AppleWorks to create "knowledge base" stacks which will allow you to link related topics with simple transfer directives. As a student reads one article and finds a topic of interest, a simple keystroke will take him/her to another article about that topic. A history teacher, for example, could create a file dealing with the Great Depression. As the student reads through the file, "buttons" (keys or the mouse) may be pressed to take him/her to a related segment which might provide in-depth information on Herbert Hoover, Apple Annies, or Black Tuesday. The same process can be used to take the student to a glossary or set of questions. Any number of transfers may be made and the student can always move back to the original document with the press of a key.

Classroom teachers will find that StoryWorks opens up an entire new world of applications for creative writing. Programmed instruction, lesson plans, seating charts and student records can all take on new dimensions using StoryWorks. Librarians can prepare AppleWorks files which list books according to certain categories. Students can then select a category with a single keystroke and be taken to either a sub-category or book list. They might then select a book and be shown card catalog information or actually see a brief report written by a student who has already read the book! Administrators will also find StoryWorks adds significant power to their AppleWorks applications. Office files can be created connecting student names with teacher notes or health information. Any application where alternative or branching is inherent will lend itself to use with StoryWorks.

StoryWorks requires a sense of adventure, a little curiosity, and a bit of imagination. It will work on any Apple II+, IIe, IIc, IIgs or Laser 128. StoryWorks requires a minimum of 64k of memory and one disk drive.

StoryWorks stacks are created with the AppleWorks word processor. Therefore, users who wish to create stacks will need "classic" AppleWorks (StoryWorks reads AppleWorks files created with versions 1.0 through 3.0). StoryWorks does NOT work with AppleWorks GS files.

◆ ◆ ◆

Getting Started

1. **Before doing anything else**, insert your ORIGINAL StoryWorks disk in drive 1 and turn on your computer.
2. The first screen you see will ask you to type your name. Do this carefully. This is a one-time procedure. The blinking underscore cursor behaves as the AppleWorks insert cursor does, and the <DELETE> key may be used to delete the character to the left of the cursor (II+ owners substitute <CONTROL-D> for <DELETE>). Make sure your name is spelled correctly, then press <RETURN>. The information is stored on your StoryWorks disk and you will not be able to change it. If the first screen you see says ENTER PATHNAME TO RAM DISK refer to Pathname Problems in the "Trouble Shooting" section of this manual.
3. You will then be asked for the name of your school. Carefully enter this (or your business name or home address if you'd prefer) and press <RETURN>.
4. You will now come to the Main Menu for StoryWorks. Select option 4 (Quit) and proceed to the directions below.

We Need Your Support

Backup your disk – At this point you should remove your copy of StoryWorks from the disk drive. You may use your System Utility disk or any other copy program to make a backup copy of StoryWorks. Put your original away in a safe place.

StoryWorks is not copy-protected, and may be copied for your own use. However, it is copyrighted, so copies cannot be legally given away or sold. If a friend (maybe the teacher across the hall) is interested in using StoryWorks, he/she can order it just as you

(hopefully) did. As teachers ourselves we know all the excuses educators sometimes use for sharing "just one" copy with someone else. "It's for the kids," "We'll order a legal copy when requisition time comes," etc. A tremendous amount of work has gone into creating StoryWorks. If you give copies away then you have stolen from the creator and publisher. You may make multiple copies for use ONLY in your classroom or home. If your school or district would like to use this software in more than one classroom they should contact us for details on district, building, or network licenses.

By supporting software like this, you not only obey the law and feel good about yourself, you also encourage us to continue developing friendly, easy-to-use software at a reasonable price. You support us and we'll support you!

The SOFTWARE REGISTRATION CARD which came with StoryWorks is your key to software updates. It will allow you to receive technical support, notices of new stack disks, and general information of interest to StoryWorks owners from TI&IE. **Be sure to fill it out and drop it in the mail today!**

◆ ◆ ◆

What to do NEXT

1. Insert the StoryWorks disk (the working copy you made above) in drive 1 of your computer and turn on your machine or (if your machine is already turned on) start StoryWorks by pressing <CONTROL-OPEN APPLE-RESET> You can install StoryWorks on a hard disk by copying the file STRYWRKS.SYSTEM to any directory on the hard disk. To start up StoryWorks (e.g., from a program selector), simply run the file STRYWRKS.SYSTEM. Hard disk users will not be starting StoryWorks from a boot disk.
2. You will be presented with the StoryWorks Main Menu. Notice that your name and school name have been added to the StoryWorks Main Menu.
3. You may use StoryWorks in either large text mode (30-column mode) or 80-column mode. The default setting is for large text mode. You can switch between modes only from the Main Menu. (See Control Keys Available in StoryWorks)
4. On the Main Menu, choice one, CHOOSE A NEW STORY, will be highlighted. Press <RETURN> to select this option.

5. You will be presented with a list of sample StoryWorks files. The option SELECT ANOTHER DISK will be highlighted. Since we will initially be examining a file found on the StoryWorks disk itself we will NOT select this option. If you had a data disk with StoryWorks files in drive two (or anywhere else) you would access it by using SELECT ANOTHER DISK.

6. Use the arrow keys to scroll through the list. Highlight A FIRST LOOK and press <RETURN>.

7. Your disk drive will run, and you can watch the screen as StoryWorks quickly reads the file into memory.

8. The story you selected (A FIRST LOOK) will be in your computer's memory, and the first part of the story will be displayed on your screen. Read through it and follow the on-screen commands. Come back to this documentation once you've read through A FIRST LOOK.

◆ ◆ ◆

Reading StoryWorks Files

You've now had an opportunity to read your first StoryWorks file. StoryWorks will hold only one story in your computer's memory at a time. By returning to the Main Menu (do this by pressing <ESC>) and selecting CHOOSE A NEW STORY you can read the other Sample files. You may read these now or go on to the next section of this manual for additional information about StoryWorks.

◆ ◆ ◆

Scrolling the StoryWorks Text Display

StoryWorks provides a "More text" indicator (a short horizontal line) in either the upper right hand or lower right hand corner of your screen (or both) if a segment contains more text than will fit on your screen. If this "more text" indicator is in the upper right corner of your screen it means there is more text above the window; pressing the up-arrow will scroll that text into view. If there is a "more text" indicator at the lower right corner of the display it means there is more text below the window; pressing the down-arrow key will scroll that text into view.

The text display will scroll by one line if you press one of the arrow keys. The right- and down-arrow keys will scroll the window downward (text moves upward on the screen); the left- and

up-arrow keys will scroll the window upward (text moves downward on the screen). Moving the mouse away from you is equivalent to pressing the up-arrow; moving it toward you is equivalent to pressing the down-arrow.

Scrolling can be accelerated if you hold down one of the apple keys while using the arrow keys or moving the mouse. <Apple/down-arrow> will move the window down one full screen; <apple/up-arrow> will move the window up one full screen.

CONTROL Keys Available in StoryWorks

From the StoryWorks Main Menu you have the option of using several "CONTROL keys." These are keys you use in combination with the key labeled CONTROL. To use these hold down the CONTROL KEY, then type the letter key. The <CONTROL KEY> options are:

<CONTROL- S> (sound) — to turn on or off StoryWorks sound effects. (The default setting is SOUND ON.)

<CONTROL- T> (text) — to switch between 30- (large text display) and 80-column text. This switch affects the screen display of your documents and not the StoryWorks menus. Note: switching between 80- and 30-column text modes erases any story in memory. You can "read it back into memory" by selecting CHOOSE A NEW STORY **after** switching to the desired text size. The default setting is 30-column display.

<CONTROL- D> (display score) — displays your score if you are reading a StoryWorks file created with score keeping capabilities (more on this later).

<CONTROL- C> (clear) — clears your score from memory

<CONTROL - F> (free) — displays the amount of free memory available in RAM (more on this later).

<CONTROL-RESET> will return you to the StoryWorks Main Menu from anywhere in StoryWorks. This erases the stack in memory.

<CONTROL-Z> (mouZe) this is a CONTROL key combination available to you while your are reading a StoryWorks stack. It simulates the mouse button so users who have no mouse may read stacks which have been designed to be used with a mouse. <CONTROL-Z> will have no effect if the stack has not been programmed to use the mouse button.

What *is* a StoryWorks File, Anyway?

The basic element of a StoryWorks file is called a "segment." A segment is text created in the AppleWorks word processor module. A segment may be of any length (up to about 2,700 words), and may contain many paragraphs of text. Each segment is assigned a unique number with the AppleWorks "set a marker" command. The segments do not have to be numbered consecutively; nor do the numbers have to be in ascending order (or any other order).

StoryWorks segments are linked to one another by "buttons." A button is simply one of the keys on your Apple II keyboard, or the button on the AppleMouse II (if your machine has a mouse). Each segment may be linked to any other segment through one or more buttons; that is, when any part of the first segment is being displayed, the reader may jump to a display of the second segment simply by pressing one of the buttons that link the first segment to the second.

The completed AppleWorks file, in which all of the segments and their transfer directives are saved, is called a StoryWorks "STACK." It is saved to disk as a standard AppleWorks word processor (AWP) file. The StoryWorks program disk is then used to read the stack (in either 80 or 30 columns), activate the transfers and the sound effects (StoryWorks may be silenced if the user wishes) and, if you wish, keep score. The size of a stack (number of kilobytes) is limited only by the size of your computer's memory and the limitations AppleWorks puts on the size of word processor documents. A stack may contain from 1 to 1024 segments.

If you are familiar with the AppleWorks word processor then you are well on your way to being able to create your own StoryWorks files.

Each segment begins and ends with an AppleWorks "set a marker" option. (In the AppleWorks word processor, "set a marker" options may be entered from the "Printer Options" menu, <open-apple/O>.) The number of the marker is the number of that segment. Figure A shows you the first segment (segment number 0) in the StoryWorks stack A.FIRST.LOOK

[Figure A]

————Set a Marker: 0

CONGRATULATIONS!

You've loaded your first StoryWorks file (also called a Stack).

StoryWorks stacks are created using the AppleWorks word processor. This stack will show you some of the interesting things StoryWorks will allow you to do.

Press <ESC> if you wish to return you to the Main Menu.

Press SPACEBAR to continue.

————Set a Marker: 0

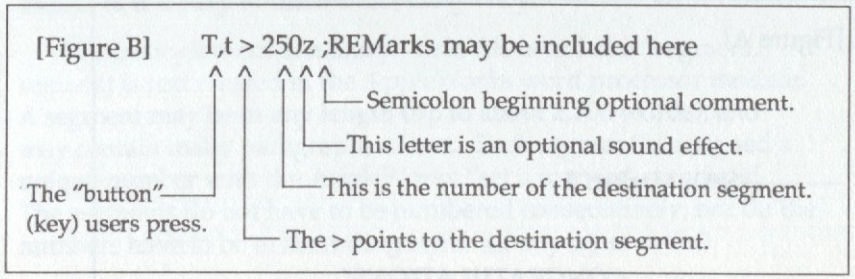
Each segment has a unique number, so that StoryWorks can find it. The segment shown in Figure A is numbered zero.

Transfer Directives

StoryWorks segments are linked to one another by "buttons." A button is simply one of the keys on your Apple II keyboard, or the button on the AppleMouse II (if your machine has one). Each segment may be linked to any other segment through one or more buttons; that is, when any part of the first segment is being displayed, the reader may jump to a display of the second segment simply by pressing one of the buttons that link the first segment to the second.

FOLLOWING EACH SEGMENT is a set of "transfer directives" that specify how the preceding segment is linked to the other segments. Each transfer directive is a short AppleWorks paragraph that specifies a set of one or more buttons, the segment number to which the buttons will send the reader, and (optionally) a code for a sound effect that will accompany the pressing of one of the buttons in the set. There may be any number of transfer directives for each segment; that is, each segment may be linked to any or all of the others in any way you wish. Each segment has random access to all of the others. For example, the transfer directive shown in figure B means,

[Figure B]



"If either the <T> or <t> key is pressed, transfer the display to segment number 250, and make sound effect Z." If you wish to include notes to yourself about the transfer directive you may do so by preceding them with a semicolon. The remark may be as long as you wish, but will terminate at the first carriage return.

◆ ◆ ◆

Creating Your First Stack

Before attempting to create your first stack, you should be sure to read the previous two sections of the manual: *What is a StoryWorks File, Anyway?* and *Transfer Directives.*

Once you have read these sections and used StoryWorks to load and read the StoryWorks stack, *A First Look*, you are ready to experiment with creating your first stack. It is assumed that users of this section are familiar with the basic procedures for loading AppleWorks and with basic AppleWorks word processor editing functions.

1. The first step in creating your first StoryWorks stack is to boot AppleWorks. Then use your StoryWorks disk as you would any AppleWorks **data** disk.

2. Load the file StackStarter to your AppleWorks desktop. This is a file (template) which has been created to assist you in constructing stacks. It consists of a prologue (the text before any markers), fifty-one sets of markers, and two standard transfer directives for each set. You will also notice some ^ (carets). These indicate AppleWorks "sticky-spaces". A sticky-space followed by a carriage return is used to insert a blank line in StoryWorks stacks. Change the name of the StackStarter file <APPLE-N> to StackOne.

3. Zoom in <APPLE-Z> so you can see embedded print commands, carriage returns, and, of critical importance to StoryWorks, **markers**.

4. Since we will only be using three segments for our first stack you will need to delete most of the StackStarter file. Move your cursor to line 39 (Set a Marker 3) and delete <APPLE-D> everything from there to the end of the file.

5. Now move your cursor to the line just after the first Marker 0 which begins with the words "You may begin your stack by... ". You can use your overstrike cursor to write over the text between the first two Marker 0's or you may wish to first delete this text. Then enter text so that you end up with the following:

```
————Set a Marker: 0
^
This is my first StoryWorks stack. There is only one thing more
fun than reading StoryWorks stacks, and that is creating them.
^
If you want to learn more about StoryWorks, press L
If you want to return to the Main Menu, press ESC.
^
————Set a Marker: 0
ESC> !; this transfer returns the user to the Main Menu
RTN> <; this transfer takes the user back to the previous segment
————Set a Marker: 1
^
————Set a Marker: 1
ESC> !
RTN> <
————Set a Marker: 2
^
————Set a Marker: 2
ESC> !
RTN> <
```

6. The text you entered between the markers is what will appear on the first screen in your stack. Notice that you have given users of your stack two alternatives, "Press L to learn more about StoryWorks" and "Press ESC to return to the Main Menu". It is usually a good idea to provide stack users with a way back to the Main Menu. The conventional method is by using the ESC key.

7. Now look at the **transfer directives** immediately following the second Marker 0. The first (ESC > !) tells StoryWorks to return to the Main Menu when the user presses the <ESC> key. The second is a "revert" transfer directive. It causes StoryWorks to

"back-up" to the segment from which you came if you press <RETURN>. The use of the revert transfer directive is explained in detail elsewhere in this manual. Since you told your stack users to "Press L to learn more about StoryWorks" you will need to add a transfer directive for the letter L (see step 8).

8. First you will need to insert a line for your transfer directive immediately after the second Marker 0. Place your cursor on the "E" in ESC (line 29). Press <RETURN>. Arrow up one line (to the line you inserted) and enter this transfer directive:

   L,l > 1a

9. The transfer directive you entered tells StoryWorks to go to the segment numbered "1" and to make the sound associated with the letter "a" (see Sound Sampler stack). Notice that both the upper and lower case versions of the letter L were included in the directive. This means that the transfer will be made if the user presses an upper case L **or** a lower case l.

10. You are now ready to enter the text for your next screen. Move to the space just after the ^ caret which follows Marker 1 (line 33). Press return and then enter the following text.

```
————Set a Marker: 1
^
"StoryWorks is a program which allows users to create their own
software. You link screens of text with transfer directives. The
screens are created with AppleWorks."
^
Press N to go to the next screen.
ESC to return to the Main Menu.
RETURN to go back one screen.
————Set a Marker: 1
ESC > !
RTN > <
```

11. You will need to add the transfer directive for N in the same way you did for the letter L in step 8. Place the directive after the second Marker 1. Enter N,n > 2

12. You are now ready to create your last screen for this stack. Place your cursor on the line after the blank line for segment 2. Type the following text: "This is the end of my first StoryWorks stack. It has been saved as an AppleWorks file. You may press <ESC> to return to the StoryWorks Main menu or <Return> to back up one screen."

13. Save your completed stack (there should be room on your StoryWorks disk or you can use another). You can now start StoryWorks and read your stack!

◆ ◆ ◆

A Close Look at a Stack

One of the best ways of learning how to create your own StoryWorks Stacks is to examine some of the Sample files on the StoryWorks disk. You can do this by using AppleWorks.

1. Boot AppleWorks (1.0 to 3.0). StoryWorks will work with any version of "classic" AppleWorks.

2. Use your StoryWorks disk as you would any AppleWorks data disk. Select Add files to the Desktop. You will see the StoryWorks files displayed (they are AppleWorks word processor files) as you would any other AppleWorks file. Select the file A.First.Look.

3. Be sure to ZOOM-IN (Apple-Z) so you can see the carriage returns, printer commands, and the location of the markers.

4. The first paragraph of A.First.Look is the "prologue" for the stack. A stack prologue is used to provide authors with notes and documentation they may need. It is especially helpful to note any changes authors can make to "customize" the stack (see the prologue for Treas.Mystery). The prologue is optional and is entered BEFORE your first **Set A Marker** command. The information in the prologue will NOT be displayed when you use StoryWorks to read your stack.

5. Notice the first **Set a Marker** command. It is numbered 0. Using number 0 for your first marker is the conventional method used in StoryWorks stacks. However, you may use any number from 0 to 254. Following the **Set a Marker** command comes the actual text of the segment. This is the information which StoryWorks will display on the screen. Finally the second **Set a Marker** command is entered (the second half of the "marker sandwich"). Notice that it has the same number (zero) as the first marker. The second marker number does not have to match the number of the first marker, but this is a convenient convention to follow. When editing a stack in Appleworks, you then may use "Find a Marker" (<OA-F>, <M>) to move unambiguously to your segment.

6. Following the text of your segment come the transfer directives. A transfer directive is made up of the "button" or key the user will press, the greater than sign > which points to the target segment, the target segment number (or symbol), an optional letter designating the sound which will accompany the transfer, and finally any optional comments the creator included (these must be preceded by a semicolon).

To summarize: a StoryWorks stack consists of an optional prologue, followed by from one to 1024 segments. Each segment is sandwiched between two "set a marker" options and is followed by a set of transfer directives that link it to other segments. Examining the sample stacks on this disk (A.First.Look, Civil.War. Quiz, Eighty.Column, Sound.Sampler, The.Civil.War, and Treas.Mystery) will help you see how to create your own stacks using the AppleWorks word processor.

◆ ◆ ◆

Special target symbols

! Using the exclamation mark (!) as a target will return a user to the main menu. Example: ESC > !

< Using the less than symbol (<) as a target (this is the StoryWorks revert symbol) will return a user to the previous (parent) segment. For example A,a > < means "Press <A> or <a> to return to the segment from which you came".

NOTE: If there are two symbols (<<), the transfer will be to the grandparent segment; three symbols (<<<) will take you to the great-grandparent segment

Note: be sure to refer to the transfer diagrams in the back of the manual

/ If the "<" is accompanied by a division symbol (/), the transfer will be to the parent (or grandparent or great grand-parent) segment plus one. This feature implements a true subroutine capability. For example A,a > /< means "Press <A> or <a> to return to the segment **following (orignal segment + 1)** the segment from which you just came".

. The period (.) is used as a target when you wish to link a set of buttons to a sound effect, but not to a new segment (i.e., "when any of these buttons is pressed, make sound effect 'x' and stay right where you are"). For example A,a >.t means "Press <A> or <a> to hear the sound for the letter t and stay here".

***** The asterisk (*) is a special character when used as a button. It will be interpreted by StoryWorks as a wildcard; that is, it represents any key (including the mouse button) that has not already been specified in a previous transfer directive for the current segment. For example: * > 12 means "press any key to go to segment 12". If you are including more than one transfer directive for a given segment the asterisk "wildcard" directive should be the LAST TRANSFER DIRECTIVE in the set. This is done to prevent the wildcard directive from "overriding" your other transfers.

If you actually wish to use the asterisk key as a button (and not a wildcard), enter ASTERISK instead.

◆ ◆ ◆

Keeping Score

+ or - Inserting the plus (+) sign just before the target segment number in a transfer directive will cause the correct score to be incremented; to cause the incorrect score to be incremented, a (-) symbol must be included. For example:

C,c > +12 Typing <C> or <c> will cause the correct score to be incremented by one, then transfer to segment 12.

W,w > -13 Typing <W> or <w> will cause the incorrect score to be incremented by one, then transfer to segment 13.

**$** From within a story the scores may be displayed by including a dollar sign ($) symbol in the destination portion of the transfer directive. This will cause the scores to be displayed, following the sound effect (if any) and score incrementing (if any), but prior to the transfer to the next segment. For example:

C,c > +$12 Typing <C> or <c> will increment the correct score, cause the scores to be displayed, then transfer to segment 12.

NOTE: Be sure to refer to the transfer diagrams in the back of this manual for additional explanations.

Special Buttons

Buttons are the keys users press to implement a transfer. They are placed at the beginning of a transfer directive. In the transfer directive [T,t > 12] the T and t keys are the buttons which take a user to segment twelve.

You can use almost any key on your keyboard as a button. By including both the lower case and upper case buttons in the set, as shown above, you may cause StoryWorks to ignore the <shift> and <caps lock> keys for alphabetic buttons.

If you wish to use the following keys as buttons you will need to use special words in your directive. Here are the words:

FOR THESE BUTTONS	USE THESE WORDS
,	COMMA
>	GREATER.THAN
space bar	SPACE
esc	ESCAPE
tab	TAB
delete	DELETE
return	RETURN
mouse button	MOUSE

StoryWorks is very forgiving when it comes to the spelling of these special codes. COMMA, for example, may be spelled any way you wish so long as it has at least two characters and the first character is a "C". This rule applies to the other special codes as well. Abbreviations such as COM, GT, SPC, ESC, DEL, RTN, MS and AS (for asterisk) are perfectly okay.

◆ ◆ ◆

Non-Buttons

There are a few keys that may NOT be used as buttons. These are: <RESET>, <CONTROL>, <SHIFT>, <CAPS LOCK>, the <OPTION>, <APPLE>, and <ARROW> keys. Otherwise, any key may be used.

Using Dot Lines

An alternative way to number each segment is to place a "dot line" at the beginning of the segment (i.e., **immediately following the "set a marker" option**). A "dot line" is a one-line AppleWorks paragraph that begins with a period. The period is followed by the segment number, like this:

.255

If a "dot line" is found, the marker number will be ignored. Since 254 is the upper limit AppleWorks sets on marker numbers you may use the dot line method if you have more than 255 segments (a StoryWorks stack may contain up to 1024 segments). You may also wish to precede each segment with a "new page" option so that when you print your AWP file (from AppleWorks) each segment will begin on a new page:

————New Page
————Set a Marker: 3
.255

Unlike "set a marker," the "dot line" approach makes the segment numbers visible on an AppleWorks printout of your file.

◆ ◆ ◆

Including Remarks in Your Directive

You may find it useful to include notes to yourself or other authors with your transfer directives. This can be done by placing a semicolon at the end of the directive. You may then enter comments (remarks). The comment may be any length, so long as it does not contain an end-of-paragraph marker (i.e., a carriage return). The carriage return ends the comment.

If you wish, the AppleWorks hanging "indent" feature may be used effectively to make lengthy comments stand out from the transfer directive itself for easy reading. Another trick to make your printouts more readable is to place an AppleWorks "new page" command immediately prior to each segment in a stack.

Separating Segment Paragraphs

To insert a blank line in a StoryWorks segment (between paragraphs) use the AppleWorks "sticky-space" command (apple-space bar) followed by a carriage return. A sticky-space/carriage return may also be used if you wish to create a "null" segment (a

segment that has zero displayable characters). Null segments can be used to create multiple-button transfers (see Tips section).

◆ ◆ ◆

Error Checking

Each time StoryWorks loads a stack into memory it checks the file for syntactical correctness. If syntax errors are found, the AWP line number in which the error occurred will be printed to the screen. If there are so many of these that the first one scrolls off the top of the screen, you will have to read the stack again. This time, when you first see error messages appear, press <ESCape> immediately to abort the read operation. Then you can see the line number of the first error (see Trouble Shooting section).

It's a good idea to build your stack piece by piece, just a few segments at a time, and test for syntax errors by reading the stack into StoryWorks. That way any errors are easy to locate and fix.

If, during the testing of a stack, you find that you forgot to provide an exit path back to the StoryWorks menu, you can always return there by pressing <CONTROL/RESET>. This will erase the stack in memory. You can get it back by selecting CHOOSE A NEW STORY and then highlighting the desired stack.

◆ ◆ ◆

Using StoryWorks with Larger RamDisks

When you start up StoryWorks it will attempt to locate its default RAM disk. This RAM disk will be used by StoryWorks to build its working copy of the current stack; therefore, the RAM disk (the working directory) must have enough free space to hold the largest stack you intend to read. As shipped, the default RAM volume pathname is /RAM/. This default location was selected because the majority of users have a portion of computer memory (auxiliary memory) named /RAM. At the StoryWorks Main Menu you can view the size of this "working directory" by pressing <CONTROL-F> (free space). On a 128K computer /RAM, when empty, has 119 free "blocks". A block is equal to .5 Kilobytes. A computer with 128K of memory will be able to handle StoryWorks stacks of about 59K. That's about twenty single-spaced pages. This is probably more than adequate for most users.

If you are using a machine with more than 128K of memory you may want to change this default setting to take advantage of your extra RAM. This will allow you to use StoryWorks stacks larger than 59K. To change the pathname of the StoryWorks "working directory" you press <ESC> as the program is loading. StoryWorks will stop and display the current working directory location (probably /RAM/). You will see the message ENTER PATHNAME TO RAM DISK. At this point you may change this to any ProDOS™ pathname (use your delete key to remove the unwanted name, II+ owners use <CONTROL-D>). If you are using a IIe with a memory expansion card in slot 5 you would change the location to /RAM5/. On a IIgs the size of the RAM disk is determined by control panel settings. The pathname will probably be /RAM5/.

With certain Applied Engineering and Checkmate Technology RAM expansion cards (e.g., RamWorks, Z-RAM, but not RamFactor) you may have to run an initialization program so as to build the RAMdisk prior to launching StoryWorks. This initialization program usually will have a name such as "PRODRIVE." If you wish, you can build a StoryWorks boot disk that automatically runs PRODRIVE. Just format a blank disk (use a utility program, not *AppleWorks*) as volume /STORYWORKS, then copy to it the following files: PRODOS, BASIC.SYSTEM, PRODRIVE, and STRYWRKS.SYSTEM. Now boot up the new disk, type in the following short AppleSoft BASIC program, and save it to disk as "STARTUP" (using the command "SAVE STARTUP" as shown):

```
]NEW

]10 D$=CHR$(4)

]20 PRINT D$;"-PRODRIVE"

]30 PRINT D$;"-STRYWRKS.SYSTEM"

]SAVE STARTUP
```

Of course, the filename "PRODRIVE" must be replaced by whatever name your RAMdisk initialization program has. Now the RAMdisk will be initialized each time you boot up your /STORYWORKS disk. You will have to change the StoryWorks default RAMdisk pathname to the name of your RAMdisk volume (typically /RAM for Applied Engineering, /MRAM for Checkmate Technology).

What Pathnames May I Use?

If you are unsure of whether or not you have extra RAM (or its correct pathname) you can quickly find out by selecting "Choose A New Story" from the StoryWorks Main Menu. At the next menu select "Select Another Disk" then "Display Volume Names". StoryWorks will display the names of any on-line RAMdisks, floppy disks, or hard disks. Once you know the name of the device you wish to use for your StoryWorks "working directory" you can reboot StoryWorks, press <ESC> during the loading, and change the ProDOS pathname to one which was listed during the Display Volume Names procedure.

Owners of machines with 128K of memory who would like to use larger stacks may select a hard disk, or even a floppy disk, instead of the RAM disk for StoryWorks' temporary stack storage (working directory). Simply follow the boot procedure above (boot StoryWorks, press <ESC>, change the Pathname). The pathname you enter would be the volume name and (if you wish) any subdirectory names. Using a formatted 5.25 disk in this way will allow users to read StoryWorks stacks of about 136K. A 3.5 inch disk used in this way will allow for stacks of 796K! It should be noted that using a floppy disk in this way will cause StoryWorks to run slightly slower, since it must occasionally access the disk.

When you change the name of the StoryWorks "working directory" this information is saved on your StoryWorks program disk. It is unnecessary to change it again unless you wish to use the disk on another computer or your hardware configuration changes. You may even create different working copies of StoryWorks to work on computers with differing hardware. In this way your students should never be faced with the ENTER PATHNAME TO RAM DISK screen.

Using StoryWorks on a 64K Apple

If you own a 64K Apple (or have a room full of them!) with two disk drives you may use StoryWorks by changing the "PATHNAME TO RAM DISK" (see preceding section) to the name of the floppy disk in your second drive.

◆ ◆ ◆

Using StoryWorks on Single-Drive Systems

IIc+ or Laser 128 owners with a single 3.5" disk drive who wish to use larger stacks may follow the above procedure (see Using

StoryWorks with Larger RamDisks). Set the PATHNAME TO RAMDISK to /StoryWorks/. This will allow for stacks of up to 380K !!

You may even use StoryWorks on a single-drive (5.25") 64K machine by renaming this PATHNAME TO RAM DISK from /RAM/ to /STORYWORKS/. You will have to store your stacks on your StoryWorks disk, and you'll be limited to stacks of about 50K in size. However, StoryWorks will still be very useful even in this mode — and you'll be able to put those "useless" 64K machines to work in your classroom.

NOTE: When creating StoryWorks stacks you will need to use a computer with more than 64K to run any version of AppleWorks after version 1.3. If you are using an earlier version of AppleWorks on a 64K machine your desktop size (hence your StoryWorks stack size) will be limited to about 10K. However, StoryWorks will READ larger files and works perfectly on a 64K machine!

◆ ◆ ◆

Trouble Shooting Storyworks Error Messages

Syntax Errors

When StoryWorks reads your stack into memory it checks for errors. If it finds an error it tells you the AppleWorks line number in which it occurs. You can then use AppleWorks to examine the specific line and correct the error. SYNTAX ERRORS will be returned if you have not used the correct syntax in constructing your transfer directives.

Unexpected End of File

If you fail to include any marker in your file (or in the event that you forget to include a final marker) StoryWorks will return the error message UNEXPECTED END OF FILE.

Duplicated Seg #

If you give two segments the SAME segment number StoryWorks will return this error message. Note: You CAN use the same segment number at the beginning and end of a segment. However, you may not name two different segments with the same number.

Segment is Too Large

If your segment exceeds 2,700 words in length StoryWorks will return this error message. The easiest remedy is to divide your segment into smaller segments by inserting markers and transfer directives.

Volume (Disk) is Full

If StoryWorks returns the message VOLUME (DISK) IS FULL it means your stack is too large to use with your current RAM disk (the working directory used by StoryWorks). The solution to this problem really depends upon the hardware you are using. If you have a 128K IIe or IIc you can increase the size of the stacks you can read by making a floppy disk serve as your RAM disk (StoryWorks working directory). Do this by formatting a disk (you might name it FLOPPYRAM) with AppleWorks. Place this disk in drive 2. Boot StoryWorks and press ESC immediately. You will see a screen which says ENTER PATHNAME TO RAM DISK. Delete (use your delete key, II+ owners use <CONTROL-D>) whatever name is currently displayed and enter /FLOPPYRAM/. Now when you use StoryWorks you will need to have the FLOPPYRAM disk in drive 2. You can remove your StoryWorks program disk (once the program has completely loaded) from drive 1 and place your stack disk in that drive. Using a floppy disk as your RAM disk will allow you to read stacks of about 136K in size.

Segment Transfer Problems

Transfer always returns to first screen – If your transfer directive returns you to the "top of your stack" (the first screen in your stack) when you don't want it to, you have probably omitted a target number from the directive or you used a target number which hasn't been assigned to a segment.

Press the button and nothing happens – If you press the correct button key for your directive and nothing happens you have probably neglected to include BOTH the lower and upper case letters in your transfer directive. Note: You do not have to include both upper and lower case letters. However, if you don't include both, the stack user must press the correct (upper or lower case) letter.

Sound Problems

No sound from StoryWorks – At the Main Menu press <CONTROL-S> to toggle between sound on and sound off (the default is for sound on). Make sure you have included a sound key in your transfer directive. If you are using an Apple IIc, IIc+ or Laser 128 make sure the volume is turned up. If you are using an Apple IIgs make sure the volume is turned up (this is done from the control panel).

Sounds seem high pitched and too short – If you are using an Apple IIgs or a computer with an accelerator card or chip, your system speed may cause the StoryWorks sounds to be distorted. Set your system speed to normal if you wish to change this.

Score Keeping Problems

Inaccurate scores – Remember that StoryWorks scores remain in memory until you clear them or read a new stack. If a student reads a stack which keeps score it will be necessary to clear that score (<CONTROL-C> at the Main Menu) before the next student uses the same stack. If this is not done, the second student's scores are simply added to those of the first student.

Students changing scores – You can cause StoryWorks to display a student's score from within the stack by including a $ in your transfer directive. If you don't want a student to be able to return to a question to change his or her score then don't provide a button (or make it known to the student) which allows him or her to return to a given question or set of questions.

Scrolling Problems

Stack users aren't seeing directions – You may have critical instructions at the end of a segment. If your segment is too large for one screen your stack users must know that they need to scroll up or down to read it all. Be sure to tell them (you might want to include this information on your first screen) that the horizontal line in the upper right and lower right corners of the screen means MORE TEXT and that they need to scroll up or down (until the line disappears) to read it all.

Pathname Problems

If the first screen you see when you boot StoryWorks says, "ENTER PATHNAME TO RAM DISK" you are probably using StoryWorks on a 64K machine or you are using a copy of StoryWorks which has been configured for a different computer or for use with a floppy disk that isn't in your second drive. Enter /STORYWORKS/ as the pathname, then proceed to the main menu. From the StoryWorks' Main Menu you may determine what ProDOS™ pathnames are available for your use (see Using StoryWorks With Larger RamDisks).

◆ ◆ ◆

StoryWorks Tips

AppleWorks Power

When constructing your stacks, remember that ALL the editing power of AppleWorks is at your disposal. Modifying existing stacks is often less work than creating new stacks from scratch. Inserting markers and transfer directives in existing AppleWorks documents will transform your current AppleWorks word processor files into

StoryWorks stacks. AppleWorks functions such as Search and Replace, Find, Move, Delete, and Copy are all useful when doing this. Remember, you can copy markers in the same way you copy text. Use the clipboard to copy frequently used transfer directives.

Two Computers

It is very useful to be able to work on your stack and experiment with reading it into StoryWorks at the same time. If you have more than one computer available, run Apple Works on one and StoryWorks on the other. In this way, you can create and edit in AppleWorks and then immediately test your creation with StoryWorks.

Owners of computers with expanded memory may wish to copy both StoryWorks and AppleWorks to their RAM disk. This will facilitate moving from StoryWorks to AppleWorks.

Templates

Stack templates such as StackStarter (on your StoryWorks program disk) are very useful. When you develop a particularly useful stack you might want to consider saving the "shell" (the markers and transfer directives minus your text) as a separate file. In this way you might develop standard shells for true/false stacks, multiple choice stacks, or word completion stacks.

Multiple Letter or Full Word Transfers

It is possible to require a user to enter more than one letter or button to complete a transfer. This is accomplished by using "null segments" (segments containing only a sticky space). In essence the user presses a button and is transferred to a segment exactly like the one from which he came or with no text in it. A second (or third or more) keypress then takes him or her to the final destination. Examine the sample file WordAnswers on your StoryWorks disk to see how this is accomplished.

Don't Lose Your Way

One of the challenges any stack creator faces is remembering where he/she is in a stack and what paths go to which segments. It is very helpful to create a flowchart of your stack as you create it. You can do this with a pencil and paper or use the flowchart samples that came with this manual. Another useful method is to create a file card for each segment. Put key segment information (the first line of text, segment number, transfer directive) on the card. See TI&IE disk 61/62 for a set of macros which quickly create flowcharts on your AppleWorks spreadsheet.

Revert Directives

Revert directives are extremely useful. If you re-enter a stack from the main menu, and if the topmost segment of the stack has a button (typically the <RETURN> key) programmed for a single revert, then pressing that button will take you back to the segment from which you exited to the main menu. This lets you pick up where you left off, in case you accidentally exit the stack.

Provide an Escape Route

You usually want your stack users to be able to return to the Main Menu whenever they wish. A standard method is to include a transfer directive to the Main Menu with each segment. This is accomplished by programming the ESC key to ALWAYS return a user to the Main Menu. The transfer directive is ESC > !

Provide a Step Back

You will sometimes want to provide your stack users with a way to back up one step. This is usually done by programming the RETURN key for a single revert. The transfer directive is RETURN > <

◆ ◆ ◆

Additional Educational Applications

StoryWorks Applications

The usefulness of StoryWorks is incredible. Your own tutorials, quizzes, and knowledge base stacks will quickly enhance your classroom software library.

Some teachers have found it helpful to use StoryWorks with an LCD monitor and an overhead projector to assist in sharing notes with students. This allows you to move through your notes in an endless number of ways, depending upon the needs of your students.

Create stacks which hold poetry or prose. Provide buttons which will give readers definitions for new words or phrases. You can also provide a "glossary button" in each segment which will allow your users to jump to a list of terms and their definitions.

Include a "hint key" in some segments. The hint key can provide users with additional information about a question.

If you are using StoryWorks with students to enhance creative writing be sure to let them first explore some of the current published works in the twist-a-plot style. These include the *Choose Your Own Adventure* ™ and *Time machine*™ series from Bantam Books, *Find Your Fate*™ series from Ballantine Books, and *The Choice is Yours*™ from Troll Associates.

LIMITED SITE LICENSE

Registered owners of StoryWorks may make up to, but no more than, thirty-five (35) copies of the StoryWorks program disk for use in a classroom. These copies may be used in only one classroom at a time and must have the registered owner's name or school name displayed on the Main Menu/Title screen. Before using StoryWorks with systems, computer networks, or Apple emulations, you must first obtain a special license from Teachers' Idea & Information Exchange.

Those wishing to use StoryWorks in more than one classroom or building should contact TI&IE, P.O. Box 6229, Lincoln NE 68506, (402) 483-6987 for building or district license information.

◆ ◆ ◆

Teachers' Idea & Information Exchange

StoryWorks users around the world are sharing their stacks (and other AppleWorks files) through the Teachers' Idea & Information Exchange. We'd like to hear about how you use StoryWorks. If you would like to share your stacks through the monthly Teachers' Idea & Information Exchange send them in today! If your stacks are selected for inclusion on one of our monthly disks, we'll give you a FOUR disk TI&IE subscription or you may select FOUR of our back disks (a $30.00 value). If you'd like to subscribe to TI&IE send us a copy of the form found on the inside of the back cover of this manual and your check. We'll enter your subscription and send you our free catalog and index disk right away!

Send StoryWorks stacks or other AppleWorks files you would like to have considered for inclusion on a TI&IE disk to:

TI&IE
P.O. Box 6229
Lincoln, NE 68506

Acknowledgements

The StoryWorks author Robert C. Moore and Teachers' Idea and Information Exchange wish to express their sincere thanks to the following individuals for their suggestions and help during the development and testing of StoryWorks: David Chesebrough, Leetsdale PA; J. Ernest Cooper, Lathrup Village MI; Dorothy Cordes, Girard IL; Peter Crosta, Nutley NJ; Mark Fegan, Peru NE; Phil Johnson, Lincoln NE; Barry McDonald, Forestburgh NY; David Moorhead, Norris NE; Robert Netro, Canton OH; Oliver Roosevelt III, Fairforest SC; Janet Sheppard, Paragon IN.

◆ ◆ ◆

Customer Support Information

If you have questions or problems (or would just like to have a friendly conversation), you can contact the TI&IE Technical Support Staff for expert assistance.

Before calling, you should check this instruction manual to see if it contains the information you need. Write down a complete description of the problem and the version number and serial number (inside the front cover of your manual) of your StoryWorks disk. It may also be helpful to print out your stack or call from a phone which allows you to be at your computer. While we will be happy to talk with anyone, we can only answer technical questions which come from registered owners. If you have not returned your product registration card, do so immediately.

Technical Support:

(402)-483-6987
9:00 am to 5:00 pm, weekdays (central standard time)
or you can write:

TI&IE
Tech Support
P.O. Box 6229
Lincoln NE 68506

Appendix A

Figure 1, "The Revert Operator," is a diagram that illustrates how the StoryWorks revert operator works. To understand the diagram, assume that segments A, B, C, and D are part of the normal flow of the stack, and that segments E, F, and G are part of a side excursion that may be "called" from several places in the stack. For example, the side excursion may be a small reinforcement routine that says, "Good Job!" and gives the student an opportunity to view his/her scores so far. The normal flow of the stack is A -> B-> C -> D, but assume that in segment B the student selects the button that transfers to the side excursion (segment E). A single revert from segment E (the topmost level of the side excursion) will return the display to the calling segment (segment B). Similarly, a double revert from the second level, or a triple revert from the third level, will return control to the calling segment.

Even if there are no side excursions, the revert operator permits the student to "back up" to the previous segment, no matter what that segment was. For example, if the student has traversed A -> B-> C -> D, and if the <RETURN> key (for example) is programmed for a single revert (<) in each of those segments, then three presses of <RETURN> will take the student back through -> C -> B -> A.

If you re-enter a stack from the main menu, and if the topmost segment of the stack has a button (typically the <RETURN> key) programmed for a single revert, then pressing that button will take you back to the segment from which you exited to the main menu. This lets you pick up where you left off, in case you accidentally exit the stack.

This revert feature is especially useful when implementing hypertext knowledge bases. Let's say you're in a knowledge base about the Civil War, and you select a button that sends you to a segment that gives more information about Abraham Lincoln. Then while reading about Lincoln, you press a button to see expanded information about Lincoln's assassination. If you revert back to the segment on Lincoln, you will see the exact screen display that you left when you pressed the button for assassination information. Similarly, when you then revert back to the Civil War segment you will return to the exact page in that segment from which you originally came.

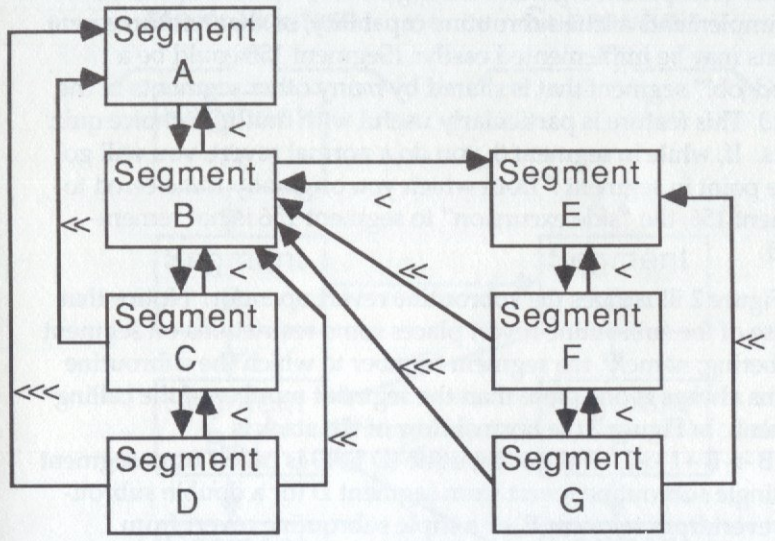


Figure 1: The REVERT Operator

If the "<" is accompanied by a "/" symbol (e.g., "/<"), the transfer will be to the top of the parent (or grandparent or great-grandparent) segment number plus one. That is, if you transferred to segment 156 from segment 7 and the selected transfer directive target in segment 156 is "/<", you will transfer to the top of segment 8. (The parent segment was segment 7; 7 plus 1 equals 8.) This implements a true subroutine capability, so that reinforcement screens may be implemented easily. (Segment 156 could be a "Good job!" segment that is shared by many other segments in the stack.) This feature is particularly useful with multiple-choice quiz stacks. If, while in segment 8, you do a normal revert, you will go to the point in segment 7 from which you originally transferred to segment 156; the "side excursion" to segment 156 is not remembered.

Figure 2 illustrates the subroutine revert operator. Notice that the use of the subroutine revert places some restrictions on segment numbering; namely, the segment number to which the subroutine returns always is one more than the segment number of the calling segment. In Figure 2 the normal flow of the stack is A -> B -> B +1 ->C. If the subroutine (D E, F) is called from segment B, a single subroutine revert from segment D (or a double subroutine revert from segment E, or a triple subroutine revert from segment F) will return the student to segment B+1; that is, the subroutine returns to the segment whose number immediately follows the number of the segment from which the subroutine was called (assuming successive numbering). Notice that successively numbered segments do not have to be located near each other in the stack; segment numbering within a stack is completely unrestricted.

If, after returning to segment B+1 from the subroutine, the student moves on to segment C, then does a normal revert three times, he/she will traverse -> B+1 -> B -> A; the side excursion through the subroutine is not remembered by StoryWorks. Any modifications to the scores that were made by the subroutine will be remembered, however. Reverts of any kind to not affect the scores.

You can "back up" only so much. If the user attempts to revert too many times, the display will return to the very first (topmost) segment of the stack. If the parent segment is segment 1023 and the transfer destination is "/<", the display will return to the topmost segment of the stack (there is no segment 1024). Any time you attempt to transfer to a segment number that has not been defined in the stack, the display will go to the top of the first segment in the stack.

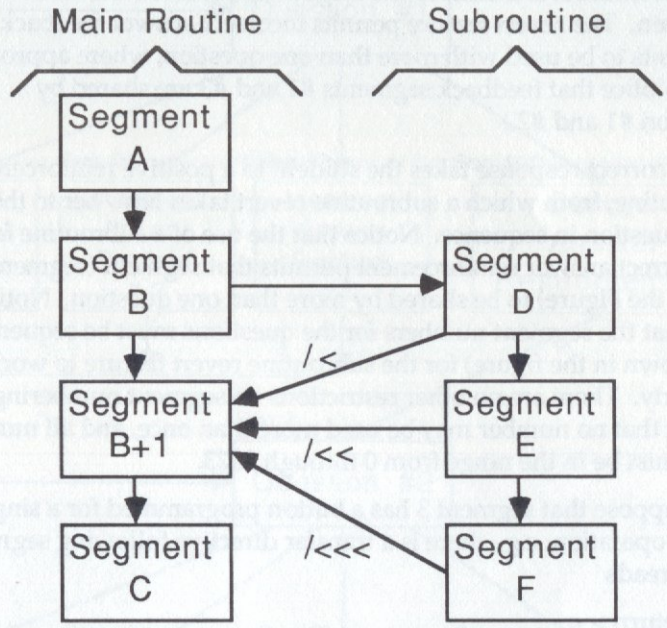


Figure 2: The SUB-REVERT Operator

Figure 3 illustrates the stack flow for a typical multiple-choice quiz. Following an initial instruction segment, the student is presented with multiple-choice questions. Question #1 is in segment 1, question #2 is in segment 2, and so on. An incorrect response takes the student to an appropriate feedback segment. From there control reverts back to the original question so the student can try again. The revert feature permits incorrect answer feedback segments to be used with more than one question, where appropriate. (Notice that feedback segments #1 and #3 are shared by question #1 and #2.)

A correct response takes the student to a positive reinforcement subroutine, from which a subroutine revert takes him/her to the next question in sequence. Notice that the use of a subroutine for the correct answer reinforcement permits that segment (segment 100 in the Figure) to be shared by more than one question. Notice also that the segment numbers for the questions must be sequential (as shown in the figure) for the subroutine revert feature to work properly. There are no other restrictions on segment numbering, except that no number may be used more than once, and all numbers must be in the range from 0 through 1023.

Suppose that segment 3 has a button programmed for a single revert operation; e.g., there is a transfer directive following segment 3 that reads

Return > <

(This revert path is not shown in Figure 3.)

What will happen when the student is viewing segment 3 (Question #3) and he/she presses <RETURN>? The revert will take the student back to segment 2 (Question #2), because the subroutine excursion to segment 100 is not remembered by StoryWorks. Similarly, a single revert from segment 2 will take the student back to Question #1.

"Backing up" does not take you back through any revert or subroutine revert paths you may have traversed. For a multiple-choice quiz such as that shown in Figure 3, this means that you can "back up" cleanly to the previous question in the quiz (if the stack is so programmed) to try it again.

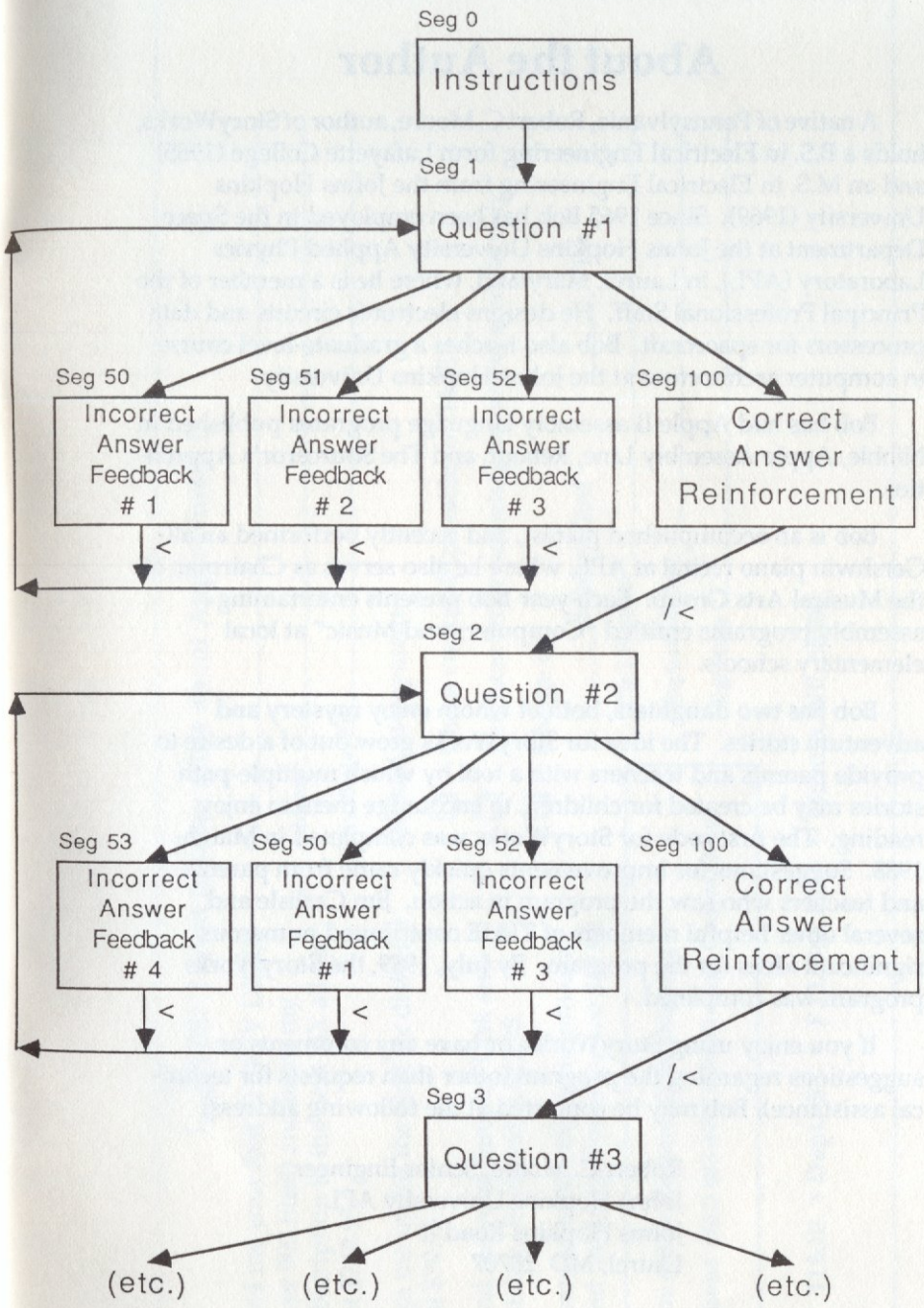


Figure 3: Stack Flow Diagram for Multiple-Choice Quiz

About the Author

A native of Pennsylvania, Robert C. Moore, author of StoryWorks, holds a B.S. in Electrical Engineering form Lafayette College (1965) and an M.S. in Electrical Engineering from the Johns Hopkins University (1969). Since 1965 Bob has been employed in the Space Department at the Johns Hopkins University Applied Physics Laboratory (APL), in Laurel, Maryland, where he is a member of the Principal Professional Staff. He designs electronic circuits and data processors for spacecraft. Bob also teaches a graduate-level course in computer architecture at the Johns Hopkins University.

Bob has had Apple II assembly language programs published in Nibble, Apple Assembly Line, Reboot, and The Sourceror's Apprentice.

Bob is an accomplished pianist, and recently performed an all-Gershwin piano recital at APL, where he also serves as Chairman of the Musical Arts Group. Each year Bob presents entertaining assembly programs entitled "Computers and Music" at local elementary schools.

Bob has two daughters, both of whom enjoy mystery and adventure stories. The idea for StoryWorks grew out of a desire to provide parents and teachers with a tool by which multiple-path stories may be created for children, to encourage them to enjoy reading. The first code for StoryWorks was completed in March, 1988. Suggestions for improvements quickly came from parents and teachers who saw the program in action. Jim Carlisle and several other helpful members of TI&IE contributed numerous significant ideas for the program. By July, 1989, the StoryWorks program was completed.

If you enjoy using StoryWorks, or have any comments or suggestions regarding the program (other than requests for technical assistance), Bob may be contacted at the following address:

Robert C. Moore, Senior Engineer
Johns Hopkins University APL
Johns Hopkins Road
Laurel, MD 20707

STORYWORKS QUICK REFERENCE CARD

Control Keys Available in StoryWorks

Available from the StoryWorks Main Menu:

<CONTROL- S> (sound) — turns on or off StoryWorks sound effects.

<CONTROL- T> (text) — switches between 30 (large text display) and 80 column text.

<CONTROL- D> (display score) — displays score if the StoryWorks file was created with score keeping capabilities.

<CONTROL- C> (clear) — clears a score from memory.

<CONTROL- F> (free) — displays the amount of free memory available in RAM.

<CONTROL-RESET> — will return you to the StoryWorks Main menu from anywhere in StoryWorks.

<CONTROL-Z> — (mouZe) simulates the mouse button, available while reading a stack.

Special target symbols

! Returns the user to the StoryWorks Main Menu. Example: ESC > !

< The StoryWorks revert symbol will return a user to the previous (parent) segment. Example A,a > <

/ The "<" may be accompanied by a division symbol (/). The transfer will be to the parent segment plus one. Example A,a > /<

. Links a set of buttons to a sound effect, but not to a new segment. Example A,a >.t

* The "wildcard" symbol. Stands for any key not already specified in the current set of transfer directives. Example * > 10

Keeping Score

+ or - These symbols are added to transfer directives to implement scoring. For example:

C,c > +12 will cause the correct score to be incremented by one.

W,w > -13 will cause the incorrect score to be incremented by one.

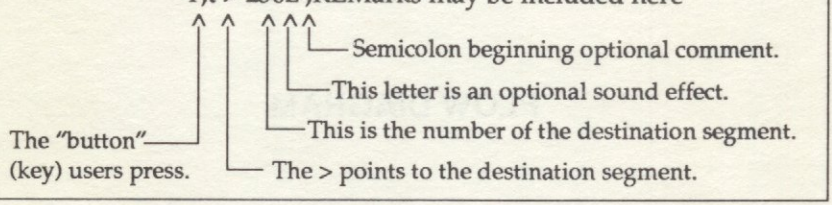
$ From within a story the scores may be displayed by including a dollar sign ($) symbol in the destination portion of the transfer directive.

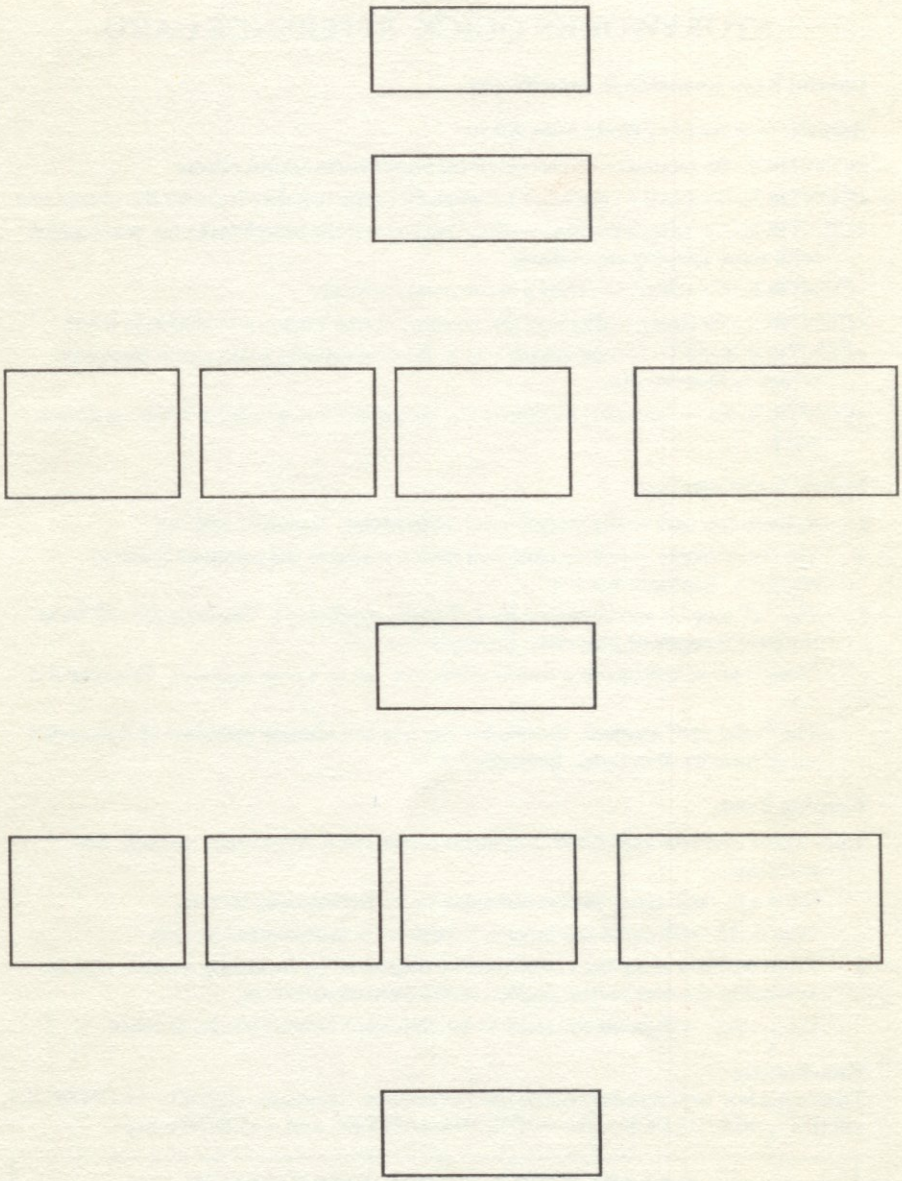
C,c > +$12 will cause the score to be displayed before transfer is made.

Non-Buttons

There are a few keys that may not be used as buttons. These are: <RESET>, <CONTROL>, <SHIFT>, <CAPS LOCK>, the <OPTION> <APPLE>, and <ARROW> keys.

SAMPLE TRANSFER DIRECTIVE





FLOW DIAGRAM