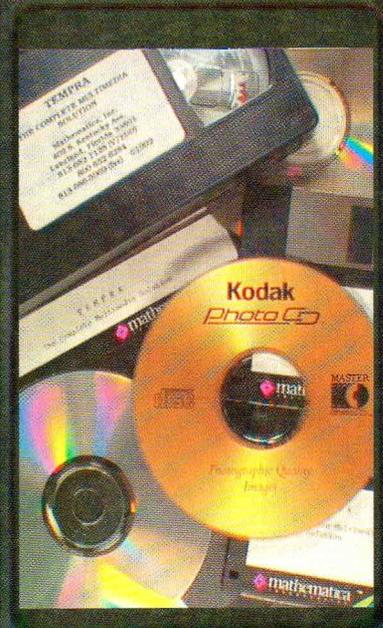
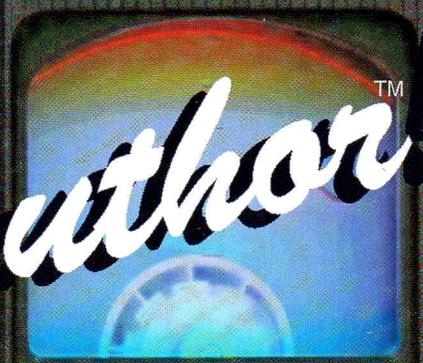


# TEMPRA MEDIA

*Author!*<sup>TM</sup>

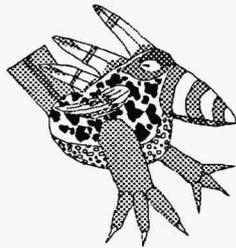
MULTIMEDIA PUBLISHING  
VIDEO PRODUCTION • CBT  
INTERACTIVE KIOSKS •  
DEMOS • ENTERTAINMENT  
COURSEWARE • TRAINING  
DATABASE • EDUCATION  
VIDEO • PRESENTATIONS  
MULTIMEDIA • KIOSKS  
CBT • TRADESHOWS  
ENTERTAINMENT •



*Reference Manual*

# TEMPRAMEDIA

# *Author*<sup>TM</sup>



Reference  
and Tutorial  
Guide

**First Edition**  
November 1992

**Copyright**

TEMPRA Media Author software created for Mathematica, Inc. by V\_Graph Inc.

Copyright © V\_Graph Inc. 1991-1992. Programmed by Trip Denton.

Graphic User Interface Copyright © 1991-1992 by Mathematica, Inc. Interface designed by Bob Schneider and Kyle Miller. All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form by any means without the written permission of Mathematica, Inc.

**Software License**

Your license agreement with Mathematica, Inc., which is included in this product, specifies the permitted and prohibited uses of the product. Any unauthorized duplication or use of TEMPRA Media Author in whole or in part, in print, or in any other storage and retrieval system is forbidden.

**Trademarks**

TEMPRA, TEMPRA PRO, TEMPRA GIF, TEMPRA SHOW, TEMPRA Media Author, TEMPRA TURBO ANIMATOR, all TEMPRA logos, VideoCAD, VideoCAD logo, Mprint, Mcapture, Mshow, Cmap, Mathematica and the Mathematica logo are trademarks of Mathematica, Inc., all rights reserved. The following trademarks are reserved by their owners: Animator, Animator Pro, AutoCAD, FLC, and Studio 3D by Autodesk, Inc.; Artist by Control Systems; Cardinal SNAPplus DESKTOP VIDEO ADAPTER by Cardinal Technologies, Inc. ComputerEyes by Digital Vision; Elographics by Elographics, Inc.; IBM and Info Window by International Business Machines Corp.; Illuminator by Matrox Electronics, Ltd.; MicroKey by Video Associates, Inc.; MS-DOS, PC Paintbrush, and Windows by Microsoft Corp.; Pagemaker, TIFF by Aldus Corp.; PaintJet, DeskJet, HP, LaserJet by Hewlett-Packard Co.; Paradise By Paradise Systems, Inc., a Western Digital Company; PC Paint and PCX by ZSoft Corp; Penny, Pepper, Pro 1280 and SGT by Number Nine Computer Corp.; PiCLink by Giddings & Lewis, Inc.; PixelMaster, ScanMaster by Howtek Corp.; Seiko by Seiko Instruments, Inc.; Sony, Vbox, Vdeck and ViSCA by Sony Corp.; Sound Blaster and Video Blaster by Creative Labs Inc.; Summagraphics by Summagraphics Corp.; TARGA, TARGA+, TGA, VISTA, NUVISTA+, VideoVGA by Truevision, Inc.; Tektouch by Graphics Technology Company; The Microtouch Screen by Micro Touch Systems; Ventura Publisher by Xerox Corp;

Genus Microprogramming, Inc.'s PCX Programmers's Tool Kit™ © Genus Microprogramming, Inc. 1988, 1989 were used as aids in development. Genus Microprogramming, Inc.'s PCX Effects © Genus Microprogramming, Inc. 1988, 1989 were used as aids in development. The Flilib library used in this product © 1990 Dancing Flame, Mill Valley, CA. All rights reserved. TOUCAN2.FLI file © 1991-1992 Digitech, Lakeland, FL. Printer drivers and support documentation licensed from Autumn Hill Software. Portions © copyright of Voyetra Technologies 1989-1992.

All other products are trademarks of their respective owners.

**Credits**

First edition produced by Martha Watkins, David Blewett, Joan Davies, Chris Murray, Rob Morris, Erin Whittier, Gary Dunn, Allen Downard, Derek Hodges, and Trip Denton.

**Front Cover Art**

Judy Frierman, Joan Davies, and Jennifer Kleman.  
Toucan Road © T.P. Speer, 1989

**Mathematica, Inc.**

402 S. Kentucky Avenue, Suite 210  
Lakeland, Florida 33801 USA  
813-682-1128 (phone)  
813-686-5969 (fax)

# Contents

---

## **Chapter 1**

Introduction .....	1-1
TEMPRA Toucans .....	1-3
Other Uses .....	1-4
System Requirements .....	1-6
Technical Support .....	1-7
Installation .....	1-7
Environmental Variables .....	1-11
Windows 3.0/3.1 Users .....	1-11
Using the Mouse .....	1-12
Using the Keyboard .....	1-12

## **Chapter 2**

Menu Conventions .....	2-1
Terminology .....	2-1
Graphic User Interface .....	2-2
Making Events .....	2-12
Building a Sample Line .....	2-13
Distributable Presentations .....	2-15

## **Chapter 3**

The Workform .....	3-1
Menu Entry .....	3-1
Keyboard Entry .....	3-2
Using Blocks of Events .....	3-3

## **Chapter 4**

The Common Fields .....	4-1
Type .....	4-1
Label and Next .....	4-1
Wait .....	4-3
Sp/Dp (Speed/Display) .....	4-4

## **Chapter 5**

The Event Types .....	5-1
-----------------------	-----

## **Chapter 6**

An Audio Event .....	6-1
Type .....	6-1
Other Fields .....	6-3
Chapter Tutorial .....	6-3

## **Chapter 7**

A BColor Event .....	7-1
Type .....	7-1
Method .....	7-1
Wipes Demo .....	7-1
Wipes Menu .....	7-2
Sp/Dp .....	7-3
Parameters .....	7-3
Other Fields .....	7-4
Chapter Example .....	7-4

---

## **Chapter 8**

A Button Event .....	8-1
Type .....	8-1
File .....	8-1
Method .....	8-2
Parameters .....	8-2
Next .....	8-3
Other Fields .....	8-4
Chapter Tutorial .....	8-4

## **Chapter 9**

A CD-ROM Event .....	9-1
Type .....	9-1
Method .....	9-1
Parameters .....	9-3
Other Fields .....	9-3
Chapter Example .....	9-4

## **Chapter 10**

A Chain Event .....	10-1
Type .....	10-1
File .....	10-2
Wait .....	10-2
Other Fields .....	10-2
Chapter Example .....	10-2

## **Chapter 11**

A DBMS Event .....	11-1
Type .....	11-1
File .....	11-1
Method .....	11-2
Parameters .....	11-8
Database Examples .....	11-16
Other Fields .....	11-18

## **Chapter 12**

A Dir Event .....	12-1
Type .....	12-1
File .....	12-1
Method .....	12-2
Sp/Dp .....	12-2
Parameters .....	12-3
Other Fields .....	12-3
Chapter Example .....	12-3

## **Chapter 13**

A Draw Event .....	13-1
Type .....	13-1
Method .....	13-1
Parameters .....	13-2
Other Fields .....	13-3
Chapter Example .....	13-4

---

## **Chapter 14**

80Butn Event .....	14-1
Type .....	14-1
Method .....	14-1
Sp/Dp .....	14-2
Parameters .....	14-2
Next .....	14-3
Other Fields .....	14-3
Chapter Example .....	14-4

## **Chapter 15**

80Menu Event .....	15-1
Type .....	15-1
Method .....	15-1
Sp/Dp .....	15-2
Parameters .....	15-2
Next .....	15-3
Other Fields .....	15-3
Chapter Example .....	15-4

## **Chapter 16**

80Mode Event .....	16-1
Type .....	16-1
Method .....	16-1
Other Fields .....	16-2
Chapter Example .....	16-2

## **Chapter 17**

80Text Event .....	17-1
Type .....	17-1
File .....	17-1
Sp/Dp .....	17-1
Parameters .....	17-2
Other Fields .....	17-3
Chapter Example .....	17-3

## **Chapter 18**

80Win Event .....	18-1
Type .....	18-1
File .....	18-1
Sp/Dp .....	18-2
Parameters .....	18-2
Other Fields .....	18-2
Chapter Example .....	18-2

## **Chapter 19**

A Global Event .....	19-1
Type .....	19-1
File .....	19-1
Method .....	19-2
Parameters .....	19-4
Other Fields .....	19-5
Chapter Example .....	19-5

---

## **Chapter 20**

A Gosub Event .....	20-1
Type .....	20-1
File .....	20-2
Next .....	20-2
Other Fields .....	20-2
Chapter Example .....	20-3

## **Chapter 21**

An IfThen Event.....	21-1
Type .....	21-1
Parameters .....	21-1
Next .....	21-4
Other Fields .....	21-4
Chapter Example .....	21-5

## **Chapter 22**

A Menu Event .....	22-1
Type .....	22-1
Method .....	22-1
Parameters .....	22-2
Next .....	22-3
Other Fields .....	22-4
Chapter Tutorial .....	22-4

## **Chapter 23**

A Midi Event.....	23-1
Type .....	23-1
File .....	23-1
Method .....	23-2
Other Fields .....	23-2
Chapter Example .....	23-2

## **Chapter 24**

A Music Event .....	24-1
Type .....	24-2
File .....	24-2
Method .....	24-2
Parameters .....	24-4
Other Fields .....	24-6
Chapter Example .....	24-6

## **Chapter 25**

A PLComm Event .....	25-1
Type .....	25-1
Method .....	25-2
Next .....	25-6
Chapter Example .....	25-6

---

## **Chapter 26**

A Screen Event .....	26-1
Type .....	26-1
File .....	26-1
Method .....	26-2
Parameters .....	26-2
Other Fields .....	26-3
Chapter Tutorial .....	26-3

## **Chapter 27**

A Sprite Event .....	27-1
Type .....	27-1
File .....	27-1
Method .....	27-2
Parameters .....	27-2
Other Fields .....	27-3
Special Features of Sprites .....	27-3
Chapter Tutorial .....	27-4

## **Chapter 28**

A System Event .....	28-1
Type .....	28-1
Method .....	28-1
Parameters .....	28-2
Other Fields .....	28-2
Chapter Example .....	28-2

## **Chapter 29**

A Text Event .....	29-1
Type .....	29-1
File .....	29-1
Method .....	29-2
Sp/Dp .....	29-2
Parameters .....	29-2
Other Fields .....	29-4
Chapter Tutorial .....	29-5

## **Chapter 30**

A Title Event .....	30-1
Type .....	30-1
Method .....	30-1
Parameters .....	30-2
Other Fields .....	30-3
Chapter Tutorial .....	30-4

## **Chapter 31**

A TmOut Event .....	31-1
Type .....	31-1
Method .....	31-2
Wait .....	31-4
Other Fields .....	31-4
Chapter Example .....	31-4

---

## **Chapter 32**

A Track Event .....	32-1
Type .....	32-1
Method .....	32-1
Other Fields .....	32-2
Chapter Example .....	32-2

## **Chapter 33**

A Var Event .....	33-1
Type .....	33-2
Method .....	33-2
Sp/Dp .....	33-3
Parameters .....	33-4
Next .....	33-13
Extended Capability .....	33-13
Reserved Names .....	33-14
Using Variables	
Successfully .....	33-16
Using Variables to	
Get to Files .....	33-16
Reading & Writing with	
Text Files .....	33-16
Series of Variables .....	33-17
Seeing the Values of	
Variables .....	33-17
Using the Read Method .....	33-17
Using the Split Method .....	33-18
Using the Write Method .....	33-19

## **Chapter 34**

A VCR Event.....	34-1
Type .....	34-2
File .....	34-2
Method .....	34-2
Parameters .....	34-4
Wait .....	34-4
Common Fields .....	34-4
Video Editing .....	34-6
Chapter Examples .....	34-6

## **Chapter 35**

A Video Event .....	35-1
Chapter Sequence .....	35-2
Video Display .....	35-2
Animation .....	35-9
Motion Video .....	35-15
Infowindow .....	35-19
Chapter Sequence Tutorial ...	35-21

## **Chapter 36**

A Vdisc Event .....	36-1
Type .....	36-1
Method .....	36-2
Parameters .....	36-3
Chapter Example .....	36-4

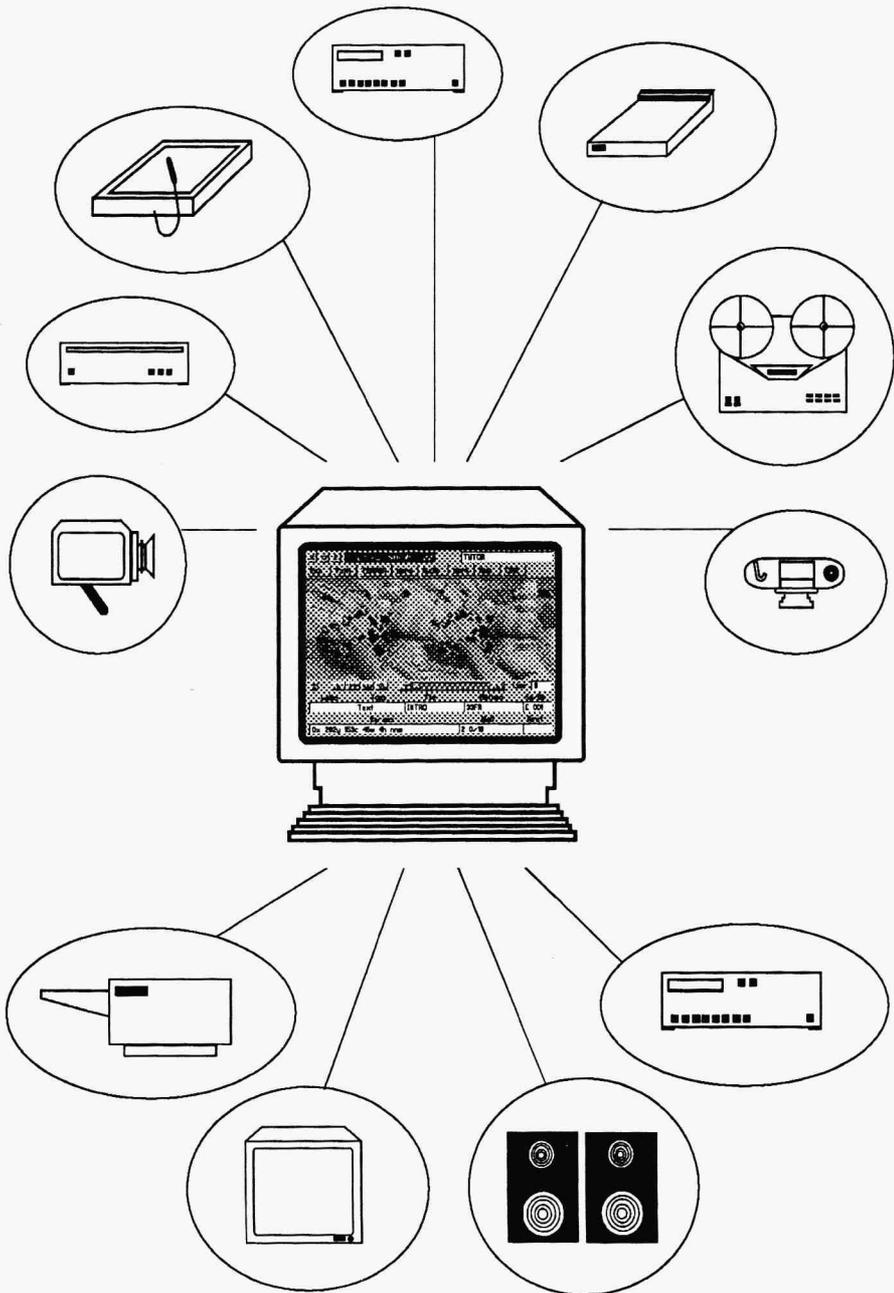
---

<b>Appendix A</b>	
Keyboard Functions .....	A-1
<b>Appendix B</b>	
TEXTEDIT Information .....	B-1
<b>Appendix C</b>	
Using Serial I/O .....	C-1
<b>Appendix D</b>	
Creating Music.....	D-1
<b>Appendix E</b>	
PWin Utility .....	E-1
<b>Appendix F</b>	
Keyboard Return Codes .....	F-1
<b>Appendix G</b>	
Font Selections.....	G-1
<b>Appendix H</b>	
Technical Information .....	H-1
CONFIG.SYS & AUTOEXEC.BAT .....	H-1
Configuration Program .....	H-2
Start Up Options .....	H-9
MIDI Event Information .....	H-10
Touch Screen Information .....	H-11
Environmental Variables .....	H-15
Troubleshooting .....	H-18

<b>Appendix I</b>	
TEMPRA SHOW Tutorial .....	I-1
Introduction .....	I-1
Using the Tutorial .....	I-3
Tutorial Structure .....	I-3
Starting the Tutorial .....	I-3
<b>Appendix J</b>	
TEMPRA Media Author Tutorial ..	J-1
Tutorial Objectives.....	J-2
Tutorial Conventions .....	J-4
Getting Started .....	J-5

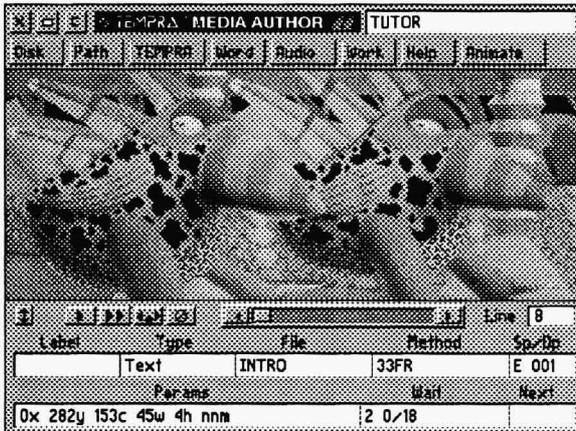
---

# TEMPRA Media Author



***The Ultimate Multimedia and Imaging Solution***

---



## Chapter 1

# Introduction

**Welcome to the world of multimedia.** Multimedia is an ill-defined word for a fascinating new way to communicate. Before you right now is a system with the power to change minds, to bring joy, to amuse and amaze. Multimedia is more than slide shows with sound, more than video computing, more than flashy animations! It is the best way to reach through the noise of modern life and really touch other peoples' lives.

We provide the tools, you add the rest. The tools: a paint, animation, and authoring system that puts you in the vanguard of multimedia. We made this software with you in mind. We have built tools and functions into software that you can use to make practically any message you want, and all without worrying about how we do it.

The great thing about this exciting new medium is the diversity of uses people put it to. Some we never expected. We have people using TEMpra SHOW, the forerunner of TEMpra Media Author, for such things as automatic slide shows from home-made videotapes, computer dating, tombstone sales, high-rise building security, factory automation, toy marketing, professional interactive resumes, cable television advertising, promo tape production, and, of course, demo disks, real estate databases, computer-based training, medical information, and trade shows...just to name a few.

The question is not “What can I do with this?” but rather, “How do I get it to do what I want?”

You may be interested to know that the software was designed by an artist and a filmmaker. The intent has always been to provide satisfaction for the creative drives within everyone, without spending a lot of money. In the process, TEMPRA SHOW and TEMPRA Media Author were influenced by the users of the software, as they started to tell us what they needed.

Basically, TEMPRA Media Author is a program that organizes your files, puts them into a sequence, and plays them back to you. These files are image files, animation files, sound files, and text files. We also provide access to database files. The approach has always been one of simplicity. We think it is easiest to arrange the things that you see and hear one at a time. We call these things “events.” This makes our authoring system event-driven. Do this...then that...then the other. We think it is best to use the computer for what it is good at: managing information and doing one thing at a time. In this case, the information happens to be images and sounds.

TEMPRA SHOW, established in the multimedia market as a powerful presentation and authoring system, has all the presentation features we could squeeze in. Just to name a few: you can bring in an image of practically any size and color depth at any time. Simultaneously, you can play midi music and animation, record audio, grab images, draw over one image with another, bring in video in a window, and you can send your message out on videotape or diskette at no extra cost. In short, we put all the fun into TEMPRA SHOW we could.

TEMPRA Media Author, a more advanced system, has all the presentation features of TEMPRA SHOW, plus the power of a programming language! We have added database support, variables, file reading, serial communications and video editing functions. The beauty of TEMPRA Media Author is that you do not have to learn how to write the language, because we write it out for you. All you have to do is make selections from the menus we provide and TEMPRA Media Author does the rest.

The presentations created with TEMPRA Media Author can be played back with a royalty-free runtime executable module called MARUN. You can also output the presentations to video for tape distribution, or by using the Demo Builder, build playback disks. This manual includes instructions for creating distributable multimedia presentations. The license to distribute complete multimedia presentations is included with your purchase of TEMPRA Media Author. You or your company may make copies of the presentations created with our software and distribute them for showing on other PCs. If you use a third-party duplication service, you need an additional license. Call Mathematica Inc. for details. Under no circumstances may you make copies of the TEMPRA Media Author authoring system for distribution to others.

TEMPRA Media Author is the desktop video/multimedia presentation companion to TEMPRA PRO and TEMPRA GIF imaging software. Your presentations can contain images created from an unlimited number of sources. Of course, TEMPRA PRO and TEMPRA GIF are the ideal sources. Images can be created, scanned, or video captured in TEMPRA PRO or GIF. And TEMPRA Turbo Animator is another program in our line that compliments TEMPRA Media Author. With this program, you can create your own animation or video files for playback in TEMPRA Media Author, or touch up existing animations. TEMPRA Media Author allows you to put these images and animations into motion, add sound, and place your text messages into a final interactive presentation or desktop video production. TEMPRA Media Author is also ideal for prototyping computer-based training.

There are many different kinds of events in TEMPRA Media Author. Each one was designed with a purpose in mind. Look through Chapter 5, The Event Types, in this manual, for an introduction to the events available.

## ***TEMPRA TOUCANS***

---

The graphic included on the cover of this manual, throughout the manual, and the front of the TEMPRA Media Author box is called Toucan Road. It was painted by T.P. Speer and has become the symbol of TEMPRA products. The bird on the left is Buzz, on the right is Sharp Boy. Both have



Sharp Boy says:  
Look for me in this manual!

developed personality since working at Mathematica, Inc. They are very adaptable birds. Buzz has been scanned, photographed, rendered, animated, palette-reduced, and shifted to gray-scale. Buzz used the colors in his feathers to teach us about color cycle. When Buzz isn't being cut up, he likes cutting up with Sharp Boy, often bragging about his position as technical advisor throughout the TEMPRA PRO reference manual. There, he is always giving image editing advice to anyone who will listen.

Well, now Sharp Boy has a new role here at Mathematica, Inc. He was recruited to be the technical advisor for TEMPRA Media Author. When it comes to building and editing sequences, you'll find Sharp Boy buzzing (no pun intended) around throughout the TEMPRA Media Author manual. Whenever you see him, you know that what he says will help you enjoy TEMPRA Media Author.

## ***TEMPRA Media Author And Other Uses***

---

TEMPRA Media Author is for presenting your ideas clearly to others. It is a software program that provides you with the ability to convey your message to your audience by using images and text, as well as video and audio. No matter what you have to say, TEMPRA Media Author helps you say it. Some of the many uses for TEMPRA Media Author are described below.

**TEMPRA Media Author is used to make PC based "slide shows."** Show your image files (PCX or TGA) on the screen with many special effects using the Screen event. You can add text to the pictures, with the Text event. You can add simple animation with the Sprite event, or sound with the Audio event. You can run through the slides automatically based on time delays, or step through them manually by pressing keys. You can go forward and backward through the slides, with the [F9] and [F10] keys.

**TEMPRA Media Author is used to prototype, or mock up, other programs.** By using the interactive Button and Menu events, you can let

your users choose what they see next as 80 column text, graphics, or true-color photorealistic images. TEMPRA Media Author is powerful enough to use in prototyping interactive video training.

**TEMPRA Media Author is used for promotional diskettes.** You can send PC disks in the mail to market your product. TEMPRA Media Author can help you distribute your message quickly and powerfully with the Demo Builder.

**TEMPRA Media Author is a video titling system.** When used in combination with a VGA-to-video output board, you can record TEMPRA Media Author text and titles on videotape. The Ticker effect in the Video event lets you run your text across the screen like a ticker tape, while the scroll-up effect lets you scroll text up the screen like credits. Some video boards provide overlay functions so you can show text or animation over live or taped video. TEMPRA Media Author is a titling system with animation and audio to act as a video production environment.

**TEMPRA Media Author is a video production system.** With multiple deck Sony VISCA support, you can daisy-chain several VISCA devices together for video editing through the computer. This eliminates the need for expensive specialized video editing hardware.

**TEMPRA Media Author is used to create interactive displays.** Trade shows, kiosks, and store displays are more attractive if the user can make choices from a screen. The branching features and touch screen compatibility make this easy with TEMPRA Media Author.

**TEMPRA Media Author is a DOS shell program.** You've seen DOS menu systems for PC users who want to start programs without seeing the DOS prompt. TEMPRA Media Author can start any program and run it, if the program takes less than 500k to run. TEMPRA Media Author starts other programs from the graphics screen, so you can point to an icon for the program with the Button event and run the program with the System event.

**TEMPRA Media Author is a CBT/courseware development system.** By using variables and its activity tracking capabilities, you can use TEMPRA Media Author for building comprehensive computer-based training and courseware.

**All the uses for TEMPRA Media Author are too numerous to list here.** If you need to express an idea, TEMPRA Media Author can help you do this in the most powerful way you can imagine.

Though you may want to jump right into the program, please take the time to look through this manual to give yourself a good feel for TEMPRA Media Author and then experiment with the tutorials before you start building your own presentations.



Sharp Boy says:  
Use the tutorials as templates  
for your presentations...it will  
save you a lot of time.

Many of the chapters have short tutorials designed to familiarize you with that chapter's event type. As you progress through the manual, you will learn that you can import a template from one sequence into another using the Save and Load Block selections on the Workform Main Menu. When you select Load Block, you are given the choice of loading a saved block or a sequence. Of course, all information in the fields can be changed to fit your needs, so you can use the tutorial sequences as templates for your own presentations.

The tutorials at the end of this manual combine several event types for an extended presentation. Appendices I and J contain TEMPRA Media Author Tutorials.

## ***System Requirements***

---

To use TEMPRA Media Author, you must have:

- **An IBM compatible computer** with an Intel 80286 (IBM AT) or better processor. Performance is greatly enhanced with a 80386 or 80486 computer processor.
- We recommend using **Super VGA or better** (640x480, 256-color, 32,000-color, or 16.7 million-color) display adapters to run TEMPRA Media Author. It does, however, support EGA, and basic VGA graphics.

- At least **540K** of main memory.
- **DOS 3.0** or higher.
- About **5.5 megabytes** of disk space for TEMPRA Media Author program files, plus space for any shows you create.
- TEMPRA Media Author uses expanded memory that complies with the Lotus-Intel-Microsoft standard (LIM spec 4.0 or better). The more expanded memory you have, the faster your presentations run. If you don't have a memory manager on your computer, TEMPRA Media Author won't use the memory above one megabyte. TEMPRA Media Author should have a total of 3 megabytes. TEMPRA Media Author supports the Windows HIMEM memory program and DOS memory managers like QUEMM, EMM386, and 386MAX.



Sharp Boy says:  
For preferred technical support, become a member of our Multimedia Player Program. Call Mathematica, Inc. for details.

## ***Technical Support***

Mathematica, Inc. provides technical support from 9 a.m. to 5 p.m. (EST) on normal business days. An online HELP file is available in TEMPRA Media Author, which you may want to consult before calling technical support. Our technical support phone number is 813-682-1130. We also have a TDD for deaf and hearing impaired customers. We are not open on weekends or holidays. Mathematica, Inc. can be reached on CompuServe by typing "GO TEMPRA" or "GO MEDIAAUTHOR" or calling our CompuServe User ID #75300, 1760.

## ***Installation***

During installation, you can add up to two of your own DOS programs to be accessed from within the TEMPRA Media Author graphic menu. This allows you to enter these programs without having to exit TEMPRA Media Author.

One of these programs can be a text editor or word processor that you already use frequently. TEMPRA Media Author comes with an editor called **TEXTEDIT**, which is accessed by selecting the **Word** button on the graphic menu unless you specify another word processing program to be used.



Sharp Boy says:

I have installed TEMPRA Media Author so many times it has become routine. We made it easy for you. Just hit [Enter] when you are asked and TEMPRA Media Author will go onto your hard disk in a flash. Of course, you can change things later, if you want.

Before starting the installation, there are a couple of things you may need to know if you will be using a text editor other than the one supplied with TEMPRA Media Author, or some other DOS application.

- Be sure you know the command (filename) that executes your text editor or word processor. This is what you normally type from the DOS prompt to call up the program.
- Make a note of the directory where your text editor or word processor is located.
- Be sure you know the command (filename) that executes your other DOS application if you choose to use one from TEMPRA Media Author.
- Make a note of the directory where this DOS application is located.

The following installation procedures include instructions for substituting your editor or word processor for the one supplied with TEMPRA Media Author, and for adding another DOS program to be called from within TEMPRA Media Author. TEMPRA Media Author is installed on your computer's hard disk by using the **Install** utility supplied by Mathematica.

1. First, be sure you are logged onto your hard drive. Type **CD\** and press the [Enter] key. This puts you in the root directory.
2. Place the disk marked "**TEMPRA Media Author Disk I**" into your A: drive (or B: drive).
3. Type **A: INSTALL** (or **B: INSTALL**) and press [Enter]. This starts the installation procedure.
4. Screen 1.0 briefs you on the installation procedure. After you've read the instructions, press any key to continue. You can cancel the installation at any time by pressing the [Esc] key.
5. Screen 2.0 shows your system configuration, the DOS (Disk Operating System) version, EMM (Extended Memory Manager) version, EMM available and installed, RAM (Random Access Memory) available and free, and the input drive. Press [Enter] to continue.
6. Screen 3.0 asks on which disk drive you wish to install TEMPRA Media Author. The default is the **C: drive**. If you have more than one hard drive available, you will see it listed on the screen. Press [Enter]

- to accept the default, or move your cursor to highlight the drive on which you want TEMPRA Media Author installed, and press [Enter].
7. Screen 4.0 asks you to confirm the directory in which TEMPRA Media Author will be installed. The default is the TEMPRA directory. We recommend using this default if you use or plan to use any or all of the TEMPRA image editing programs. This allows quick transfer between the TEMPRA products. Press [Enter] to accept the default. If you want to use another directory, use your backspace key to erase what is on the screen. Then type in the name of the directory you want to use, and press [Enter].
  8. Screen 5.0 ask for the text editor you will use. The default is TEXTEDIT, which is a text editor supplied with TEMPRA Media Author. If you want to use TEXTEDIT, press [Enter] to accept the default. If you choose another text editor or word processor to use in creating files for import into TEMPRA Media Author, it must be a DOS application that supports ASCII text exports. Use your backspace key to erase what is on the screen. Type in the name that you normally type to start your text editor (users of Word Perfect would type WP). Press [Enter] to accept your selection.
  9. Screen 5.1 is related to the previous one and confirms the directory of the text editor you want to use. The default is the \TEMPRA directory, where the TEMPRA Media Author text editor is located. If you want to use TEXTEDIT, press [Enter] to accept the default. If you chose to use another text editor and entered its name above, type in the name of the directory where that editor (or word processor) is located, then press [Enter].
  10. We have provided a means for you to run existing DOS programs that you already own from within TEMPRA Media Author. In screen 6.0, you can set up this program to appear as a button on the TEMPRA Media Author graphic menu. The default is “no program.” If there is a program you want to use, type in the name of the file that executes the program. The word you type will appear on a button that you can select in TEMPRA Media Author. For instance, if you have a video capture card and want to use our TEMPRA Turbo Animator movie maker feature to bring video .FLC or .FLX files into TEMPRA Media

Author, you could set the program to include a button for **ANIMATE** on the graphic menu. If you have this feature, type **ANIMATE** in the empty box and press [Enter].

11. Screen 6.1 appears for you to enter the directory for the DOS program you want to use. TEMPRA Media Author needs this information in order to locate and execute the program. Type in the name of the directory where your program is located and press [Enter]. If you are not setting up a button in TEMPRA Media Author for an existing program, just press [Enter].
12. Screen 7.0 asks if you want to view the READ.ME file after installation. Press [Enter] to accept the default to read it. The installation process begins and you will see filenames being shown on the screen as TEMPRA Media Author extracts them to the hard disk. This process takes several minutes to complete. During the process you may be asked to insert another disk. When prompted to do this, remove the disk that is in the disk drive and replace it with the disk then press [Enter].
13. After the files have been extracted, a screen appears asking if you want your AUTOEXEC.BAT (auto execute batch) and CONFIG.SYS (configuration system) files automatically updated. Unless you're familiar with DOS commands, press the "Y" key to automatically update. If you want to change them yourself, refer to the **Technical Information** section of this manual for descriptions of the environmental variables for TEMPRA Media Author. After the AUTOEXEC.BAT and CONFIG.SYS files are updated, the READ.ME files appears. Installation is completed.
14. To start TEMPRA Media Author, first **reboot** your computer by pressing the Reset button or by pressing [Ctrl], [Alt], [Del] keys simultaneously.
15. Go to the TEMPRA directory by typing **CD\TEMPRA** and pressing [Enter]. Start TEMPRA Media Author by typing **MA** and pressing [Enter].



Sharp Boy says:  
Any changes made to your  
AUTOEXEC.BAT or  
CONFIG.SYS files will not  
upset the computing you do  
in other programs.

When you've completed the installation, place the disks in a safe location to prevent damage.

## ***Setting Environmental Variables***

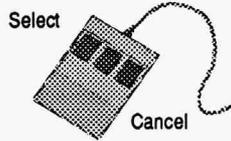
When you installed TEMPRA Media Author, you were asked if you wanted to change your AUTOEXEC.BAT and CONFIG.SYS files. If you did, your path statement was changed to include "C:\TEMPRA;". TEMPRA Media Author also added some SET statements to your AUTOEXEC.BAT file. These SET statements create DOS environment variables. TEMPRA Media Author uses the DOS environment variables to establish its configuration every time you start TEMPRA Media Author. If you want to change these variables, refer to the **Technical Information** section of this manual for descriptions of the environmental variables for TEMPRA Media Author. You may also want to refer to your DOS manual.

## ***Windows 3.0/3.1 Users***

The program disk includes two files to aid installation under Windows 3.0 and 3.1: MA.PIF and MA.ICO. MA.PIF is the program information file, which specifies the required and desired memory to run TEMPRA Media Author under Windows. MA.ICO is the TEMPRA Media Author Windows icon. You need a 386/SX PC or better to run TEMPRA Media Author under Windows.

1. Open your Windows **Program Manager**.
2. Select **File** from the command line.
3. Select **New** from the pop-up menu.
4. Select **Program Item** from the New Program Object menu.
5. In the Program Item Properties menu, type **TEMPRA Media Author** as the description and **C : \TEMPRA\MA . EXE** as the command.
6. Select the **Change Icon** button. Type **C : \TEMPRA\MA . ICO** Select OK twice to save the icon.
7. The TEMPRA Media Author icon appears in the Windows screen group. You can select the icon to start TEMPRA Media Author from Windows.

## Using the Mouse



TEMPRA Media Author is designed to work with a two- or three-button mouse. The left button is the **Select** button. It is the same as pressing [Enter] on the keyboard. The right button is the **Cancel** button. It is the same as pressing [Esc] on the keyboard. We use these terms frequently in this manual. If you are new to the computer and mouse operation, please review your mouse reference manual. We recommend proficiency at using a mouse to use this program to its full potential, although you don't have to have a mouse to use TEMPRA Media Author.



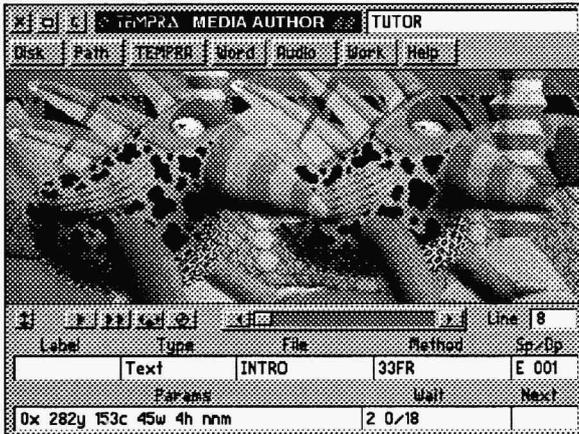
Sharp Boy says:  
"Ctrl G" lets you move  
between the Workform and  
the GUI quickly.

## Using the Keyboard

You can use certain keys on the keyboard to perform mouse tasks. The table in Appendix A of this manual describes the keys and functions they perform if pressed when you use the graphic interface (GUI) or text-based interface (Workform). Some of the most common keystrokes used are the [Esc], [F3], [F5], and [F6] keys.



- [Esc] Stops a sequence in between lines, and allows you to continue creating.
- [F3] Using the mouse, place the cursor over the field on the event controller you wish to edit and press [F3], **not** the mouse button. You can now type directly into the field the necessary information without calling on the menus for that event. Press Select or [Enter] to accept information.
- [F5] Press this key to play a single line in the sequence.
- [F6] Press this key to start and play the sequence from the current line to the end of the sequence.



## Chapter 2

# Menu Conventions

TEMPRA Media Author has an intuitive Graphic User Interface (GUI) that sets it apart from other multimedia programs. This chapter describes the terminology and conventions used in the TEMpra Media Author graphic interface.

## Terminology

Throughout this manual, conventions and functions are described in terms of making presentations. These descriptions contain words and phrases unique to the TEMpra Media Author authoring system. The following are terms and definitions that will assist you in creating presentations.

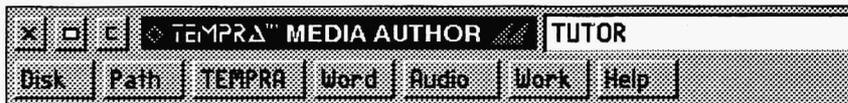
TEMPRA Media Author is a **line**-based, event-driven authoring system. An event line contains a command telling TEMpra Media Author what to do, when to do it, and how to do it. Each line is made up of an **event type** and its related **fields**. Every time you create a line, you are asked to choose from a variety of events. An event might be a graphic, audio, background color, video, text, or other selection. The related fields for each event include its file, speed and other descriptive information relevant to the event.

Several event lines together are called a **sequence**. Essentially, a sequence is your presentation. The default setting for the number of event lines in a sequence in TEMPRA Media Author is 500. If you want to create a sequence that exceeds 500 event lines, you can chain multiple sequences together so they run as one sequence. The Making Events section at the end of this chapter is a study in event line construction that illustrates TEMPRA Media Author's intuitive sequence authoring system. We recommend that you become familiar with how TEMPRA Media Author operates by reviewing this section and recreating the tutorial sequences in this manual.

## ***Graphic User Interface (GUI)***

The graphic environment of TEMPRA Media Author consists of a **Menu Bar** running across the top of your screen and an **Event Controller** running across the bottom of your screen.

### ***Menu Bar***



The Menu Bar allows you to work with disk files, exit TEMPRA Media Author, and move to other programs such as TEMPRA GIF or TEMPRA PRO. It also allows you to use the on-line help system. The following buttons (selectable areas) make up the Menu Bar.

#### **Exit**



The Exit button (or [Ctrl]-Q) takes you out of the TEMPRA Media Author program. Before you exit to DOS, you are asked if you want to save the current sequence, then you are asked to confirm the exit to DOS. Use the mouse to select your choices.

#### **Remove Menu**



The Remove Menu button removes the graphic menus from the top and bottom of the screen so you can see the entire viewing area. To return to the menus, press the Cancel button on the mouse.

## Clear



The Clear button clears the screen and changes the screen to black. Clearing does not affect your line or event in any way, but lets you sample different effects without having to replay the lines up to that point. This is very useful when you are testing various wipes or speeds on the same image.

## Title



The Title box, in the upper right of the screen, displays the filename of the current sequence you are creating or editing. By selecting the area inside the box with the mouse, you can rename your sequence. If you want to save the sequence under the new name, you must select Save Sequence from the Disk menu. Because you have renamed the sequence, your old sequence is not erased.

## Disk



The Disk button opens a menu containing various disk functions, such as loading and saving sequences, Demo Builder, Print to Video, and other disk-related functions.

## New Sequence

This menu option lets you start and name a new sequence. If you have been working on an existing sequence and select New Sequence, you are asked if you want to save the old sequence and then asked to name the new sequence. To avoid confusion and lost files, always give a new sequence a unique name.

## Load Sequence

This menu provides a list of all the sequences that are available to load and run. The files are automatically saved in the \SEQ default sub-directory unless the sequence path has been changed. The path to locate other sequences can be changed by using the Path menu.

### **Save Sequence**

This command saves your sequence to the hard drive in the \SEQ sub-directory. If a sequence with the same name already exists in the directory, you are asked if you want to write over the sequence. If you select Rename, you are prompted for a new name. The new sequence is saved to disk.

### **Rename Sequence**

This feature allows you to rename a sequence at any point. The original version is not erased unless you give it the same name.

### **View Directory**

Selecting this option lets you view the contents of the directory specified by the path you enter.

### **Delete File**

When you select this option, you are prompted to enter the path only, then press [Enter]. A list of files in the selected directory appears. Highlight the file you want to delete, press Select. A menu will pop up for you to confirm deletion. Type Y or N and press [Enter].

### **Print Sequence**

This feature allows you to output your sequence in a number of forms. When you select Print Sequence, the “Print Sequence to:” menu pops up with choices of LPT1, LPT2, File and Video. LPT1 and LPT2 allow you to print hard copy versions like a script. The print to File choice will send the print job to a file in the default directory (usually C:\TEMPRA). You can copy the print file to a floppy disk for printing from another machine. Printing to video allows you to print your sequence to a video recording device supporting the SONY VISCA protocol or a device connected to a SONY Vbox via a LANC or control L port. This provides you with another form of output for your presentation: videotape.

### **Save Picture**

This is a screen capture feature that saves any images on the screen at that moment in a PCX or TGA format. You need to provide a name for the screen file and press [Enter]. This is useful for printing images of each line

for a hard copy revision of your sequence. Save Picture will not capture other images outside of TEMPRA Media Author.

### Demo Builder

This feature allows you to put your presentations on distributable media. The viewer does not need the original TEMPRA Media Author program to view your presentation. Please refer to the section on “Creating Distributable Multimedia Presentations” in this chapter.

### Help

This option lets you access an online help file provided with the program. Help information is listed by topic.

### Quit ([Ctrl-Q])

This exits TEMPRA Media Author to DOS or Windows. You can also use the Exit button or [Ctrl-Q]. If you are in an unsaved sequence, you are asked if you want to save the sequence before exiting to DOS.

### Path



The Path button opens a menu that sets the directories TEMPRA Media Author uses to find certain files. When you install TEMPRA Media Author onto your hard disk, it creates directories to hold specific types of files. If you choose to change this directory setup, you can use the Path button to designate different paths. If you do change the paths to find your files, be sure that all of your files are located in that directory. Otherwise, TEMPRA Media Author cannot call on the images as needed. To change a path, select the disk drive, then the directory, continuing until you have the full path name. You can change the paths for files when you’re creating a sequence and save the new paths with the same name as the sequence, so when you next load the sequence, the paths will be changed so the needed files can be found.

### TEMPRA



The TEMPRA button starts the TEMPRA PRO or TEMPRA GIF image editors from within TEMPRA Media Author. Using the TEMPRA button allows you to

enter and exit TEMPRA PRO or TEMPRA GIF while running TEMPRA Media Author. Thus, the image editing and presentation programs interface directly for quick accessibility. To return to TEMPRA Media Author, select the Exit button and then exit TEMPRA.

### Word



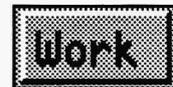
The Word button starts your DOS text editor. The installation program allowed you to identify your text editor. If you don't have a text editor, you can use the one supplied with TEMPRA Media Author called TEXTEDIT (see Appendix B for text editor information). The text editor you use must be able to edit files in ASCII format. ASCII format doesn't have codes in it to designate special format commands. MS-DOS 5.0 owners can use an editor called EDIT. EDIT produces ASCII files compatible with TEMPRA Media Author.

### Audio



The Audio button is for recording audio to disk for playback during your show and playing files back to preview them. If you use a dedicated sound card such as the Sound Blaster, you can record speech or produce music on your hard drive. (See the Audio Event chapter for more information.)

### Work ([Ctrl-G])



The Work button (or [Ctrl-G]) stands for "Workform," a text-based spreadsheet for creating and editing sequences. You can use the Workform as an alternative to the graphic interface and move back and forth between the two interfaces. The Workform screen displays the individual events in the sequence so you can see multiple event lines on the screen at one time. (The graphic menu displays events line by line.) Selecting GUI on the main menu of the Workform takes you back to the GUI screen. (See the Workform chapter for more information.)

### Help ([F1])



The Help button brings a Help index to the screen. Use the mouse to move among Help topics and select the one you



Sharp Boy says:  
Start with the GUI interface and after you have learned a bit, switch over to the Workform. You may like it. Sharp Boy uses both interfaces.

need. To access context-sensitive Help information, go to the Workform and place your cursor on the event you wish to get help information on and press the [F1] key.

## External Program



You can designate a button to appear for a program you use frequently and access that program through TEMPRA Media Author. It can be any program of your choice. This is set during installation and appears in the AUTOEXEC.BAT file as PROGX=.

## Event Controller

Label	Type	File	Method	Sp/Dp
	Text	INTRO	33FR	E 001
Params			Wait	Next
0x 282y 153c 45w 4h nnm			2 0/18	



**Sharp Boy says:**  
All fields in the Event Controller are self-prompting when you create a line starting with the Label or Type fields. However, you can reveal menus for specific fields by selecting the box below the field without affecting any other fields on a line.

The Event Controller contains the fields for the event lines that make up a sequence. Information appears in the field boxes automatically, based on your selections from the pop-up menus as you create lines.

## Menu Position



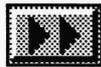
The Menu Position button changes the vertical position of the Event Controller. (It doesn't affect the Menu Bar.) When you select the Menu Position button, the Event Controller becomes a ghost box. When you move the mouse up and down, the Event Controller moves up and down. You can position the Event Controller so only certain portions of it show on the screen by pressing Select to place the ghost box.

## Play Line ([F5])



The Play Line button plays only the line displayed on the Event Controller. The menus remain on the screen while the visual events play in the display area. This allows you to review the event and make changes as necessary while creating the sequence.

### Play Sequence ([F6])



The Play Sequence button plays the sequence starting at the current line. The sequence plays in the display area between the Menu Bar and the Event Controller. To play the sequence with a full screen display and no menus visible, press [F6]. To play the sequence from the beginning, rewind to Line 1 using the Line Scroll Bar. Press [Esc] to interrupt playback of the sequence.

### Insert Line ([Ins])



The Insert Line button inserts a blank line in front of the current line and makes the current line the next one in the sequence. This pushes all subsequent lines down one. You can use the Insert Line button to make room for an event you want to add.

### Delete Line ([Del])



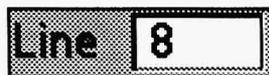
The Delete Line button deletes the current line, including the event on that line. You are given a warning that you are about to delete the line and its events. You can cancel the deletion by selecting “No” from the “Delete This Line?” pop-up menu.

### Line Scroll Bar



The Line Scroll Bar moves through lines in the sequence (similar to rewinding and fast-forwarding). Selecting on the arrow buttons on each side of the Line Scroll Bar move the sequence one line at a time. You can place the cursor inside the Line Scroll Bar to the approximate location of the line you want and press Select to move to that position. When you move through the lines in the sequence, the Event Controller displays the contents of each line. You can also “rewind” and “fast forward” using the Line Counter.

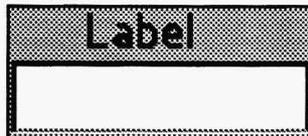
### Line Counter



The Line Counter displays the current line number. As you move the Line Scroll Bar, the Line Counter changes to reflect the number of the

current line. The events in the current line are displayed as you select each new line. You can move to a new line by selecting the Line Counter and then typing the line number you want in the GoTo menu that pops up and pressing [Enter]. Pressing Select outside of the GoTo menu returns you to Line 1.

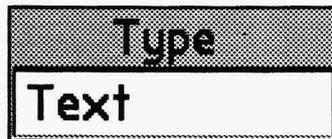
## Label



The Label field is used to identify an event or line in your show. Labels are primarily used for branching and are used in conjunction with the Next field. An example of branching can be found in the tutorial provided in this manual.

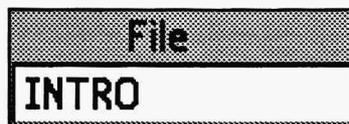
Each label must have a unique name. TEMPRA Media Author distinguishes between upper and lower case, so the words “Start” and “START” and “start” would be considered three different labels in TEMPRA Media Author.

## Type



The Type field displays the kind of event that occurs when the line is executed. The Type determines if the event is video, audio, a sprite, text, or other event. See the chapter on event types in the manual for more information.

## File



If an event like a Screen event needs a file, the menu with the possible choices appears. Highlight and then select the file you want. Some events, like BColor, do not need a file.

In this case, the file menu is automatically skipped.

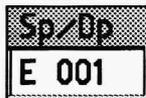
## Method



The Method field displays how an event will occur. The choices available depend on the event type. Text events have a selection of fonts as the Method, while Screen events

have a selection of wipes.

### Sp/Dp



Sp/Dp is an abbreviation for Speed and Display. This field controls either one or both of these features as they are needed by a particular event.

### Speed

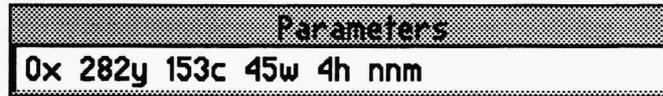
The Speed field controls the velocity at which certain events appear (it regulates the time you designate to an event). The range of speed is 1 (slowest) to 99 (fastest). For instance, the speed determines how quickly or slowly a wipe takes effect or animation is played. In a sprite event, speed controls the rate at which the sprite is moved. If your sequence is displayed on a computer that is faster than the computer used to create it, the sequence displays quicker.

### Display

The Display field determines what happens to an event in relation to a subsequent event. There are three ways of leaving the elements of your presentation on the screen:

- . *Permanent* events can be thought of as “sticking to the background.” They remain unaltered if other events are moved across or placed over them. Screen and BColor events are automatically permanent. You can only have one Screen or BColor appear at a time and a subsequent one erases the current one. You can place a scaled image over a full-screen image without erasing the full-screen image.
- . *Temporary* events reside in the “foreground,” and may be altered or removed by other events that change the foreground. They are erased by subsequent Erase after Display events and embedded into the background by subsequent Permanent events.
- . *Erase after Displaying* events also reside in the foreground and erase themselves and any Temporary event from your screen before the next event appears. You can set a Timeout for Erasing events to specify when you want them to disappear from the screen.

## Parameters



The Parameters field contains information about

the method used with a particular event, such as the color of the background, the type of display mode you're using, or the placement of video or graphics on the screen. The contents of this box change depending on the type of event. Parameters are set automatically when an event is created, but can be changed manually or with the menu.

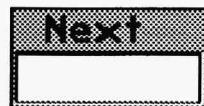
## Wait



The Wait field sets a delay between the lines in a sequence or stops the sequence for a specified time or for a key or mouse button to be pressed.

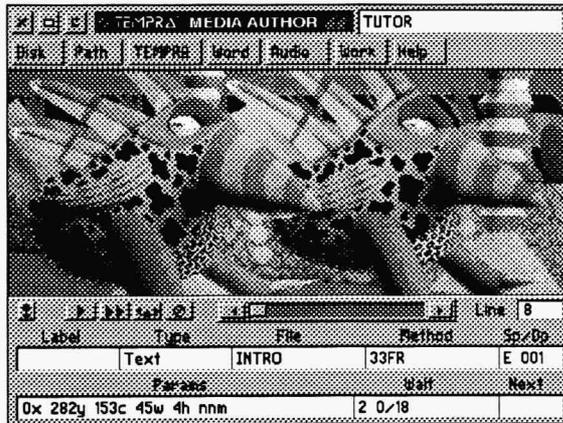
The Wait period begins when the line begins to play, so you may need to adjust the Wait depending on how long the line takes to play. You can also put a Wait on a line by itself to be sure you get the delay you want. You don't need to wait between lines and can move automatically from line to line. The default is Don't Wait.

## Next



The Next field tells TEMPRA Media Author what Label to look for and indicates the next line to be executed. The contents of the Next box correspond to the contents of the Label box. In order for TEMPRA Media Author to move to a line named in the Next box, the contents of the Label box must match exactly. If the Next box is empty, TEMPRA Media Author automatically moves to the next line in the sequence.

## Making Events

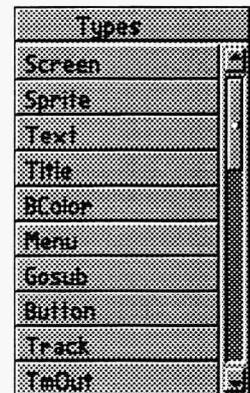


When working with the Graphic User Interface (GUI), all lines are constructed through the Event Controller. The Event Controller is the set of “buttons and sliders” on the bottom half of the screen. Using this GUI, we will construct one line to help you understand how TEMPRA Media Author does what

it does. Samples of the different line constructions are used in later sections of this manual.

TEMPRA Media Author has an intuitive GUI. It leads you through the construction of an event line automatically when you write every line. When you start a line in TEMPRA Media Author, you usually select the Type field first. It is highlighted in red. The top portion of the Type field menu appears below.

This menu asks you what type of event you want on this line. There are 31 event choices. Depending on what event you choose at this point, TEMPRA Media Author determines what other fields (File, Method, Sp/Dp, Parameters, and Wait) are needed to complete the line. On each field’s pop-up menu,



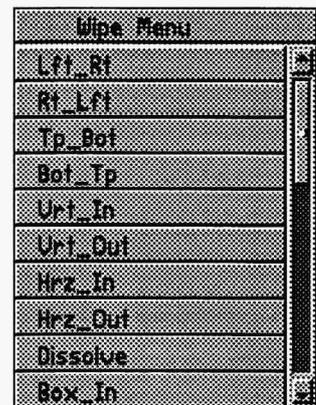
## ***Building a Sample Line in a Presentation***

When you start TEMPRA Media Author for the first time, the GUI appears and the Type field box is filled with a red highlight. When you are not branching, this is where you start most of your sequences. For this sample, we will start with the Label field and build a screen event to bring up an image.

Using the mouse, move the cursor over the first box, called Label, on the Event Controller. Press the left mouse button to select that field. A dialogue box appears on your screen. You may type a label into this box or leave it blank. For this sample line, press [Enter] on the keyboard or select on mouse to leave it blank. The Label box on the screen changes from blue to white and the Type menu appears on the screen.

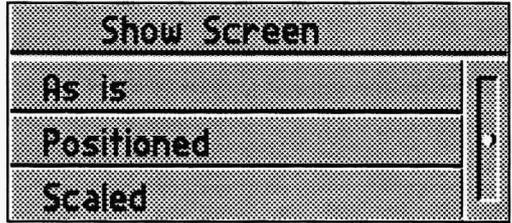
Select the Screen option to show a graphic on the screen. A list of potential image files appears for you to choose from. If you have not scanned, captured, or painted any images, this window will not list anything. For this example, we will use the tutorial images provided by Mathematica, Inc. Since the program default is .PCX the tutorial files will be listed.

Select the TUTOR2.PCX image file from the list. A wipe menu now appears. In this example, the Wipe menu determines how our image comes on to the screen. Scroll down the menu until you see Dissolve and Select it. Dissolve brings your image onto the screen in a random block pattern.



After you select the type of wipe you want to use, the Sp/Dp (Speed/Display) menu appears. From this menu, you determine the speed at which this image wipes onto the screen. The up and down arrows near the bottom of this menu make the speed slower (1) and faster (99). You can also select the window where the numbers appear and type in the speed. Because all screen events are permanent (meaning they stay on the screen until they are replaced by another visual event), the Permanent, Temporary, or Erase buttons are inactive. To move to the next menu, select "OK."

Another menu appears on the screen asking how to show the image on the screen. You can show the image “As Is,” “Positioned,” or “Scaled.” This is for the Parameters field. The Parameters field shows you the options that apply control to your selection. Select “As is.” This tells TEMPRA Media Author to place the image on the screen exactly as it would appear in a paint program at 100% size with no special positioning.



The next menu is the Wait menu. In this menu are five choices that tell TEMPRA Media Author how to proceed to the next line. Choose “Don’t Wait.”

The final menu to appear is the Next menu. This is used to control branching and is the counterpart to the Label (which we did not use). We will not use the Next field on this line, so press [Enter].



Sharp Boy says:  
Buzz gets in all the pictures  
but I'm the toucan that gets  
things moving.

You have now completed one line. On your screen is Buzz, our TEMPRA mascot. To see your line run again, clear the screen by selecting the Clear Screen button in the upper left corner of the GUI, select the line counter, go back to Line 1, and select the “Play Line” button. To see the graphic on the screen, without the menus visible, press [F6] instead of the Play Line.

Each line in TEMPRA Media Author is made the same way. The intuitiveness of the lines, events, and fields makes TEMPRA Media Author a fun, and powerful creative program to use. Enjoy!

## Creating Distributable Multimedia Presentations



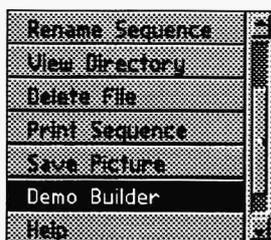
Sharp Boy says:

This is a good way to let others see your presentations even if they don't have TEMPRA Media Author. To be sure your presentations will run on any equipment your audience may have, you might want to make two levels of your demos; one for super VGA with all the extras, and one for basic computer users. Be sure to test your demos before you send them out.

You can use the Demo Builder in the Disk menu to save a sequence you create as a self-running multimedia presentation for viewing by others who don't have TEMPRA Media Author.

Here is a step-by-step example of how to use the Demo Builder.

1. To create a copy of your presentation, select the **Disk** menu on the TEMPRA Media Author Menu Bar.



2. The drop-down menu displays functions you can use in sequences. The **Demo Builder** button is not visible at first, so you need to scroll down the menu by placing the cursor in the elevator bar and pressing Select (left mouse button) or by pressing the down arrow button beneath the elevator bar. Select the bar labeled **Demo Builder**.

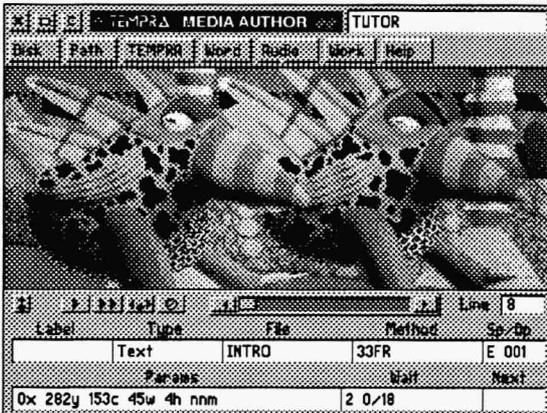
3. A menu pops up requesting the **destination** path. The default is the A: drive. You can change the destination path you want to use by typing in the path and pressing [Enter].
4. After you determine the path, the **Demo Batch File** menu pops up for you to type the name you want to give the presentation. The default demo name is "DEMO." You can change the name by typing eight characters or less in the box and pressing [Enter].
5. TEMPRA Media Author automatically creates the demo file for you. You may need to put new diskettes in the drive as the demo is being built. You will be prompted to change diskettes as necessary. The number of diskettes you need depends on the number and size of graphics files in your sequence. The demo contains a batch file to start

## Menu Conventions

---

the sequence and the files you used to build it. You can copy any sequence you have created to the demo diskettes and it can be run on another computer.

6. To run the demo file (your presentation) on another computer, copy the contents from the disk(s) built by the Demo Builder to one directory and type the name of the batch file.



## Chapter 3

# The Workform

The Workform is an alternative spreadsheet-style interface you can use to create and edit sequences. It contains the same functions as the GUI. When you look at a sequence in the Workform, you'll see it's composed of lines and columns. Each column contains a specific field. There are eight fields in the Workform. These are the same fields you see on the GUI. Most of what you need to create and edit sequences with TEMPRA Media Author can be selected from this workform. Enter the workform from the GUI by selecting **Work** with the mouse or by holding down the [Ctrl] key and pressing the "G" key. You can start TEMPRA Media Author with the Workform showing up first when you use the /W command line parameter.

## Menu Entry

You can create and edit sequences in the Workform using the drop-down menus, similar to using the menus in the GUI.

### Create from the Menus

It is possible to create and edit sequences by using the menus and entering no more than the sequence name from the keyboard. If you start in the Label field or the Type field, you are prompted automatically as you move from field to field across the Workform. If you don't want to fill in a response for the Label or Next request, just press [Enter].

### ***Edit from the Menus***

Move the field cursor over any of the fields and press Select or [Enter]. The menu and/or prompt used to create that field entry appears. Some fields may have information that is developed through responses to a series of menus or prompts. When this is the case and you want to change one step in the series, press the [Esc] key in response to the menus/prompts whose entries you don't want to change. This way you can change the end position and keep the start the same for Sprite movement, and so on. Some events have special editing menus that allow you to change only one aspect of the event, so pressing [Esc] is unnecessary.

### ***Keyboard Entry***

---

It is possible to create and edit sequences without the menus. If you are familiar with the system, you can type event types, filenames, parameters, etc. directly from the keyboard. Appendix A is a table listing all of the keyboard commands.



Sharp Boy says:  
Use the first letter of menu choices when in the Workform. The first letter gets you to an entry without having to use the down arrow key.

### ***Create from the Keyboard***

If the field is empty, position the cursor over the field and start typing. Press [Enter] when the field is finished, and you get the field cursor back. Once you are editing you can go on to the next field. You can go up and down lines. When you press [Enter] you return to the field mode, where you can use the mouse and menus.

### ***Edit from the Keyboard***

If there is information in a field, you may want to modify the field without changing everything in it. In this case, move to the field you want to edit and press [F3]. Now use the right and left arrow keys to move the small text cursor under the text you want to change, and your keyboard entries are put into the Workform. The [Home] key positions the cursor at the first character of the field. The [End] key positions the cursor at the end of the field. Any keyboard entry editing you do is in overwrite mode until you press the [Ins] key. This yields a larger cursor and puts you in the Insert mode. By pressing the [Ctrl] key and the right arrow you can go right word by word, then field by field. By pressing the [Ctrl] key and the left arrow you can go left word by word, then field by field.

## Using Blocks of Events

Using blocks allows you to insert, copy, move or delete lines of events while you are in the Workform. This is a way to make and save templates that you can use as many times as you want, in any sequence you want. As you become more familiar with TEMPRA Media Author, this is something that will be a time saver for you.



Sharp Boy says:  
Block operations are a great way to speed up creating repeating lines or creating a template for future presentations.

### Block Operations

You can copy, cut and paste part or all of your sequence to other areas of the sequence by marking blocks. You can access the block commands from the Workform by pressing [Esc]. If the **Main Menu** appears, position the cursor over **Edit** and press [Enter]. You can also save blocks from a sequence to files and load them into other sequences. Blocks are column-oriented, so you can cut and paste sections of event lines or columns from many lines.

### Marking Blocks ([F7] - [F8])

Use the [F7] key to mark the start of a block and the [F8] key to mark the end of a block. To mark the whole line, move the cursor to the **Label** field and press [F7], then move the cursor to the **Next** field and press [F8]. Notice the background for that line changes color, indicating it is “**marked.**” You may now perform one of the block operations on the block you just “**marked.**” Press the [Alt] key and “**U**” key to unmark the block.

A block cannot be used as a sequence. Blocks can be moved and copied to and from different areas of the same sequence and even from sequence to sequence, provided the sequences are in the same directory.

### Mark Block Start ([F7])

This menu selection is designed for mouse users who want to operate in the Workform without using the function keys. To designate the beginning of a block of event lines, choose the **Mark Block Start** option from the edit menu. The block then starts where the field cursor sits on the Workform. This has the same effect as using [F7] from the keyboard.

### ***Mark Block End ([F8])***

This menu selection is designed for mouse users who want to operate in the Workform without using the function keys. To designate the end of a block of event lines, choose the **Mark Block End** option from the edit menu. The block ends where the field cursor sits on the Workform. This has the same effect as using [F8] from the keyboard.

Position the cursor over the first field you want to block. Press **Cancel** using the right button of the mouse. Move the cursor down to highlight **Mark Block Start [F7]** and press **Select** using the left mouse button. Press **Select** again to reveal the Workform. Move the cursor to the last field you want to block. Press **Cancel**, then move the cursor down to highlight **Mark Block End [F8]**, and press **Select**. Notice the background for that line changes color, indicating it is “**marked.**” You may now perform one of the block operations on the block you just “marked.”

### ***Copy Block ([Alt-C])***

You can copy marked blocks throughout your sequence. The block is copied starting at the event line where the field cursor sits. It either inserts into the sequence or overwrites currently existing lines. As long as you do not unmark those lines, you can keep doing this. Use [Alt-C] or the Edit menu **Copy Block** selection to do this.

### ***Move Block ([Alt-M])***

You can move marked blocks anywhere in your sequence. Use [Alt-M] or the Edit menu **Move Block** selection to do this. The block moves starting at the event line where the field cursor sits. It either inserts its way into the sequence or overwrites currently existing fields. Once a block is moved, it has to be remarked if you want to move it again.

### ***Delete Block ([Alt-D])***

When you have a block of event lines that you want to clear from the Workform, mark them and use the [Alt-D] or the Edit menu **Delete Block** selection.

### ***Unmark Block ([Alt-U])***

Once you have a block marked, you can unmark it from the Workform with [Alt-U] or with the **Unmark Block** selection from the Edit menu.



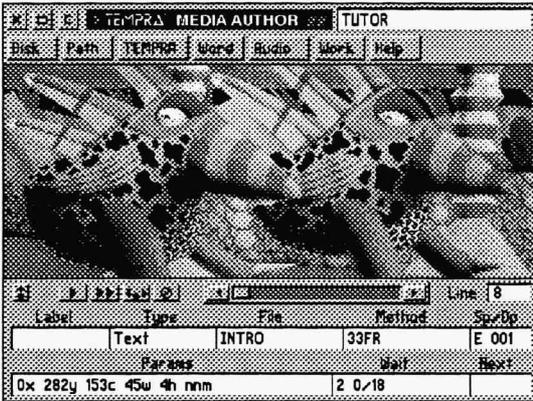
Sharp Boy says:  
Keep all of your old sequences. You can use parts of them in new sequences, deleting the lines you don't need.

### ***Load Block ([Alt-L])***

When you have saved portions of a sequence as a block, you can load them back in from disk to the current sequence. Use [Alt-L] from the Workform or the Edit menu **Load Block** selection. You are shown a directory menu that lists blocks previously saved using the **Save Block** feature. Move your cursor to highlight the block you want to load and press **Select**. They load in starting where the field cursor sits.

### ***Save Block ([Alt-S])***

When you have event lines that you want to store on disk in their own files for safety or for loading into other sequences, you can do this by saving them with the **Save Block** option. Mark them, then use [Alt-S] or the Edit menu **Save Block** selection. You will then be asked to enter a name for the block you want to save. TEMPRA Media Author saves marked blocks to disk in the same directory as sequences. However, it assigns a different extension to the filename, so you don't have to worry about loading blocks as sequences or vice versa. Sequences are saved with an **SEQ** extension, while blocks are saved with an **VBK** extension.



## Chapter 4 The Common Fields

Most of the fields presented in this chapter are used in every event line. They were first described in Chapter 2. Because most lines includes these fields, they are known as “common fields.” This chapter details their use and function. These elements are used in the same manner for every event. These commands are simple to understand and use.

### Type

Type
Text

There is a Type field in every line. This field determines what kind of event is going to take place in the presentation. The event you choose from this menu determines what other fields are relevant to this line. The Event Types chapter in this manual describes the types of events on this menu.

### Label and Next

Label	Next

Label and Next are commands used when you branch. Branching is part of an interactive presentation and is not necessary on every line. Use branching to bypass lines and execute a line that is not in numerical order. Very simply, the Label field identifies a line. This “label” is important because when you want to move to a specific line,

TEMPRA Media Author looks for the Label used to designate it. You designate this label in the Next field, the last position on the Event Controller. When a line is completed, the last thing TEMPRA Media Author looks at is the Next field. If there is a label name in it, TEMPRA Media Author searches for the line with that label and runs that line instead of proceeding to the following line in the sequence. Branching can make your presentations interactive and exciting by giving the viewer choices and a feeling of participation.

Branching commands are used in the tutorial supplied with TEMPRA Media Author. When you were asked to name the TEMPRA mascot and were given three choices, the type of reaction the program gave depended on your answer. If it was wrong, TEMPRA Media Author branched to a line that told you it was wrong and branched back to the question. If you got it right, TEMPRA Media Author branched to a line that said “You’re right!” It then continued to the next line in the sequence.

TEMPRA Media Author distinguishes between upper and lower case, so the words “Start” and “START” and “start” would be considered three different labels. The Next field you want to branch to must match the Label field exactly.

### ***Label***

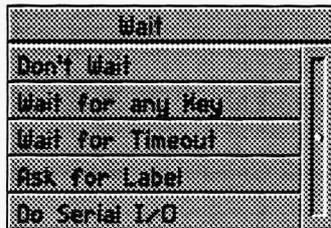
Use the Label field as a line identifier for branching. After the Label field is selected, you are prompted to type in a name. This is optional, you do not have to give every line a label, but if you ever expect to branch to this line you **must** label it. Either type in a unique name (like “Start”) and press [Enter] to create a label for the line or just press [Enter] to leave this label blank. When you use labels, **every label must be unique** or TEMPRA Media Author will not be able to find the appropriate line to branch to and the sequence will not work properly. Labels can be used on otherwise empty lines.

### ***Next***

The Next field is the counterpart command to Label. The Next field tells TEMPRA Media Author what line to go to next. That is why it comes at the end of each line. Unless you are branching, you can leave this field blank. TEMPRA Media Author automatically goes on to the next line in numerical order.

If you are branching, put the label of the line you want to branch to in the Next field. When the line has completed running, it immediately jumps to the line in the sequence with that label. This is why it is so important to give each line a unique label. If you do not do this, and you have two lines with the same label, the sequence will fail because of conflicting line identifications. Nexts can be put on otherwise empty lines.

## ***Wait***



Another common field in TEMPra Media Author is the Wait field. This field controls the time delay between lines. There are five possible options when you reach the Wait field: Don't Wait, Wait for any Key, Wait for Timeout, Ask for Label and Do Serial I/O. Waits can be put on otherwise empty lines.

### ***Don't Wait***

By selecting this option, you are telling TEMPra Media Author to proceed to the next appropriate line. There will be no delay in the sequence. Don't Wait is also the default setting for this field. By not selecting anything when you get to this field, you are actually selecting "Don't Wait." Press [Enter] to accept default.

### ***Wait For Any Key***

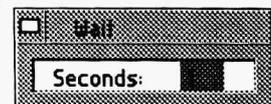
When you select this option, you are telling TEMPra Media Author not to go to the next line until the user hits any key on the keyboard. It is a good idea to prompt the viewer with a Title event just before this line. Once a key is pressed, or the left mouse button selected, the sequence will proceed to the next appropriate line. Good for creating a slide show where the keyboard or mouse acts as the advancer.



Sharp Boy says:  
Computers keep time in 18's of a second. That's why you get a choice of 18th of a second when doing a Wait for Timeout.

### ***Wait For Timeout***

This option stops the sequence for a set amount of time. When you select this option, a menu appears and you are asked to enter the amount of time you want to pass before it moves on to the next appropri-



ate line. This scheduled delay can be very useful for keeping certain graphics on the screen for guaranteed periods of time. You can break out of a timeout with a key press. The Timeout wait begins as soon as the event starts to play. If you have an event that takes 3 seconds to run, and you have a 2 second timeout in the Wait field, in essence, you will have no timeout. You can place a Wait for Timeout on the next event line to keep your image or text on the screen. Good for Erasing events.

### ***Ask for Label***

If during your presentation, you need to branch out to a specific line and do not want to use a button event to do so, use this Wait option. This is a special feature that allows you to enter a label name while the show is running and immediately branch to the event that has the label you entered. When events with Ask for Label in the Wait field are executed, a prompt appears in the upper left of the screen asking you to type in the Label. Entering the Label name branches the sequence to the line with the same Label and the sequence runs from that line. If you do not want to branch when an Ask for Label Event executes, press the [Enter] key and the sequence continues.

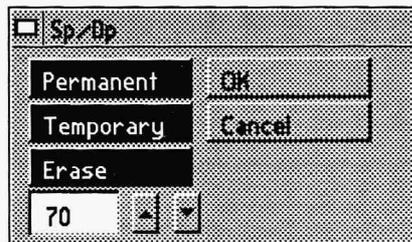
### ***Do Serial I/O***

Serial I/O, or Input and Output through the serial ports 1 and 2, can be accomplished by selecting “Do Serial I/O” from the Wait menu. This feature can also be used to send data through the parallel ports. See Appendix C, “Using Serial I/O,” in this manual for more information.



Sharp Boy says:  
TEMPRA Media Author remembers the last speed you set for an event in a sequence. This makes creating a sequence easier. You can always go back and edit the speed to synchronize your presentation.

### ***SP/DP (Speed/Display)***



Another common field is SP/DP (speed/display). Depending on the event type, you will be asked how fast you want something to happen and how you want it displayed.

## ***Speed***

When you are asked to select a speed, a menu appears with a prompt. By entering a number 1 through 99 (1 being the slowest, 99 the fastest) you will determine how fast a screen wipe happens, for example. You can select in the white box on the SP/DP menu and type in the number, press Select, then select OK. You can also select on the arrow buttons to move the numbers up or down. The speed you select does not increase or decrease the speed of your computer. Delays caused by your hardware are not improved by using a faster Speed setting. If your sequence is displayed on a computer that is faster than the computer used to create it, the sequence runs quicker.

## ***Display***

The Display field determines what happens to an event in relation to the event chosen. There are three ways of displaying the elements of your presentation on the screen:

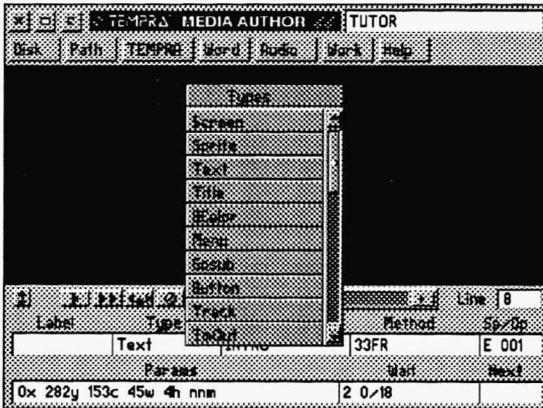
*Permanent* events can be thought of as “sticking to the background.” They remain unaltered if other events are moved across or placed over them. Screen and BColor events are automatically permanent. You can only have one Screen or BColor appear at a time and a subsequent one erases the current one. You can place a scaled image over a full-screen image without wiping the full-screen image off the screen.

*Temporary* events reside in the “foreground” and may be altered or removed by other events that change the foreground. They are erased by subsequent Erase after Display events and embedded into the background by subsequent Permanent events.

*Erase after Displaying* events also reside in the foreground and erase themselves and any Temporary event from your screen before the next event appears.

Some events require you to make a decision on only Speed or Display, while others require decisions on both.

The Display option can be used on an otherwise empty line.



## Chapter 5 The Event Types

### Types

This field determines what sort of event will happen in your presentation. Your choice of event in the Type field automatically determines the other fields necessary to complete the event line.

The types of events you can select from this menu are Audio, BColor, Button, CD-ROM, Chain, DBMS, Dir, Draw, 80Butn, 80Menu, 80Mode, 80Text, 80Win, Global, Gosub, IfThen, Menu, Midi, Music, Screen, Sprite, System, Text, Title, TmOut, Track, Var, VCR, Video and Vdisc. While the menu does not appear in alphabetical order when it first pops up, you can alphabetize it by pressing [Alt] S.

For the most part, the name of the event is indicative of what will actually happen on the screen when the line runs. A Screen event shows an image on the screen, a Music event plays music, an Audio event plays sounds or voices, etc.

This chapter briefly describes the types of events you can create in TEMPRA Media Author. For more information on creating a particular event, refer to the chapter in the manual that details it.

### ***Audio***

Audio events let you play or record sounds onto the hard disk using any supported audio card.

### ***BColor***

BColor events create background colors that wipe on the screen. Your choice of colors ranges from 16 to 16 million, depending on the display mode you have selected.

### ***Button***

Button events let you create user interactivity through branching. If the viewer is given the choice of seeing one of many possible answers to a question, a selection prompts the system to branch to the material chosen. By using the Next and Label features, it is possible to branch to anywhere from anywhere within a sequence.

### ***CD-ROM***

CD-ROM events let you play sound from standard CD-ROMs. The controls provided in TEMPRA Media Author allow you to position the CD-ROM head to the correct track and then play the music starting at that location. You can also select the range of music on the CD you want to play. This provides complete control over the starting and ending points for your music.

### ***Chain***

Chain events connect one sequence to another. Since a single sequence can contain a maximum of 500 lines, Chain events let you create presentations any length you need by chaining several sequences together.

### ***DBMS***

This event allows you to use a database in conjunction with your presentation. You can also make a database with TEMPRA Media Author. You can read, write, and add to an existing DBase III compatible DBS database file and use the information in the fields in your presentation.

***Dir***

This event allows you to bring up directory menus for your audience to use. When they select a file from the directory, the name is automatically placed into the reserved variable, \$\_file. Wild cards can be used. Paths from the .PTH file can be used to point to the directory you want.

***Draw***

Draw events let you create basic shapes, like circles or squares, either filled or unfilled. There is also a freehand tool. You can use the shapes to contain text, graphics, or other image files. You can also use the Draw event to create your own wipes.

***80Butn - 80 Column Button***

80Butn events are the same as Button events, except in an 80 column format.

***80Menu - 80 Column Menu***

80Menu events are the same as regular Menu events, except in an 80 column format.

***80Mode - 80 Column Mode***

80Mode events switch the display from graphics mode to 80 column text mode.

***80Text - 80 Column Text***

80Text events are the same as regular Text events, except in an 80 column format. No font selection is available.

***80Win - 80 Column Window***

80Win events are much like sprites, except they use text characters rather than graphics to be moved around on the screen.

***Global***

Global events set the conditions for the entire sequence and reset previous Global events. You can override the global settings in some events. For instance, you can have a Global event set the default font to one of the many

**KST** fonts. You can override the default font temporarily with a font choice in a Text event without changing the default font.

### ***Gosub***

Subroutines produce efficient applications. As a multimedia creator, you often find that you want to do the exact same thing a number of times, at a number of different places in your presentation. With the Gosub event, you can do this and still have only one copy of the event lines that does the repetitive work. All you have to do is “call” this series of lines to work from wherever you want the work done. TEMPRA Media Author will go to the subroutine, the subroutine will do the job, and then TEMPRA Media Author returns to where the routine was called. Since you can call subroutines in remote sequences, you can use the work you have done before in your current project or even have a sequence of only subroutines that are called from other sequences.

### ***IfThen***

The IfThen event is used to judge the value of variables, as well as choices taken and stored in the answer track. The result of the judgment determines the next action to be taken. The next action could be a branch to a label or the execution of the following line. Think of it this way: “IF so and so occurs, THEN jump to label-name in Next field, ELSE do what is on the next line.”

### ***Menu***

Menu events allow the viewer to select options through menus, specified by you, and branch to other parts of the sequence. This is a variation of the Button event.

### ***Midi***

Midi events let you play back audio created as Midi files (you can’t record Midi in TEMPRA Media Author). In order to play Midi files, you will need an appropriate driver for your sound board. See the READ.ME file, on-line Help file, and the Technical Information Appendix in this manual for more information.

## ***Music***

Music events control sound from the speaker found in every PC. No special hardware is needed to create Music events. You can create music using the keys on the keyboard as piano keys and save as ASCII format with a .MUS extension. See the Creating Music Appendix in this manual for help in creating your own music files.

## ***PLComm***

PLComm events read values from programmable logic controllers and branch to designated events as the result of tests performed on those values.

## ***Screen***

When you select the Type field, the first selection on the Type menu is the Screen event. Screen events show images that are created in TEMPRA PRO or GIF or other paint programs. Other things can happen simultaneously while this image is on the screen, like music, voiceovers, text or titles, and even other Screen events. Screens can be brought in as large as the resolution of the screen, scaled to a size you specify, or positioned on the screen. There are also a number of ways they can be brought on to the screen using various wipe effects. You can also determine how long images stay on the screen. A Screen event remains on your computer monitor until another Screen or BColor (Background Color) event replaces it.

## ***Sprite***

A Sprite is a small picture that can “float” over other pictures or stay in one place. You can draw an image in a paint program, like TEMPRA PRO, save it as a clipping, then use it as a Sprite, much like the “pointing hand” in the tutorial. Sprites are used when you want to point to something directly or show motion. When using a Sprite to show motion, you designate a starting point and an ending point. The Sprite then “floats” from point to point across the screen.

## ***System***

System events allow you to use DOS programs, batch files, or DOS commands during your sequence, often without affecting the visible screen.

### ***Text***

Text events display text in any graphic mode, font, or color over the image or with a frame. The text must be created with a text editor in a non-document mode (ASCII text file) with a suffix .TXT.

### ***Title***

A Title event is a display of a word or words not exceeding 9 characters. For example, you can show the command [Enter] for a press any key option. All the attributes applied to Text events also apply to Title events except that Titles are entered right in TEMPRA Media Author when you create the Title line, and are not existing ASCII files.

### ***TmOut***

The Time Out event is useful when you are using multimedia in kiosks, trade show displays, adventure games, simulations, or computer-based training. You can create internal branches based on time. **“If no key is pressed in xx:xx amount of time, then do A. If a key is pressed in xx:xx amount of time, then do B. If the time of day is xx:xx:xx, then do C.”** You can branch or jump when a timeout occurs, or you can run a subroutine.

### ***Track***

Answer tracking lets you use the viewers' choices from named menus and touch screens during button events. The Track event also allows the computer to remember the values of all the variables. You can save and load the answer track as a text file for later analysis. Technically speaking, you can use this event to save on variable declarations, because the chosen answers are always recorded and you do not have to make a scoring variable to hold a special value for each selection. You can, however, use the track during IfThen events for internal branching, or you can assign values to declared variables based on the contents of the answer track.

### ***Var***

The Variable event is used to declare variables that can then be modified, displayed, or entered from the keyboard. They can be read and written to text files. Variables can be numbers or character strings. Once the variables have values, they can be used in a database, in the IfThen events, saved to or loaded

from files. Var files can be given names from String variables. Variables can also be used in the Type, File, Method, Sp/Dp, Parameters, or Wait fields. By assigning values to the variables, you can get one line to do just about anything.

### ***VCR***

Used in conjunction with a Video event, this event lets you bring recorded video into your presentations with the SONY VISCA protocol. With a SONY Vdeck, your computer can control access to any Hi-8 footage segments. With a SONY Vbox, your computer can control any SONY video device (Hi-8, 8mm, or VHS) with the VISCA or Control L protocol. Camcorders, VCRs and high-quality video decks are yours on call for presentation video. Multiple deck control is available. You can daisy-chain up to seven VISCA decks.

### ***Video***

The Video event includes a number of selections that perform diverse tasks, from setting your display mode to showing animation and video.

**Video display** events include setting the display mode, fading images in and out, loading different color palettes, and setting a viewport on the screen. These functions control the view you get on the screen.

**Animation** events allow you to bring animation, created in popular animation programs, to a sequence. You can also animate text and colors on the screen.

**Motion video** events allow you to play live or taped video in a window within your sequence if you have the necessary hardware. In TEMPRA Media Author, the insert edit capability is available.

### ***Vdisc***

The Vdisc event is used to operate videodisc players with TEMPRA Media Author. The players that can be operated are the Sony 2000, the Pioneer 6000 and 8000. With this feature, you can search for frames, play motion video, play audio independent of the video, and more, depending on what the videodisc player can do.

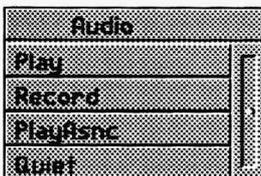


## Chapter 6

# An AudioEvent

Audio events let you play or record sounds on the hard disk using a Sound Blaster card or any other supported card. Supported cards are listed in the Technical Information Appendix of this manual under Audio Board Type. If you have a sound card other than those listed, it must be able to use Voyetra drivers. The commands in the Audio event line are the same commands as those in the Audio button menu on the Menu Bar.

### Type



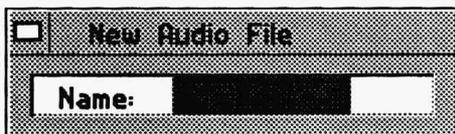
This event allows you to record and play sound files during a sequence. Once you select an audio event, a menu appears with four choices.

### Play

A menu of available audio files appears. Depending on your audio board and what boards and drivers you've selected in the CONFIGMA, only the files with the proper extensions (i.e. WAV or VOC) will appear. Select a file. From this point, TEMPRA Media Author moves on to the common elements. Be sure your audio files are in the directory selected in the path statement under the Path button. The default is \TEMPRA\AUD.

### ***Record***

The primary use of this feature is to provide you with a resource for your own audio files. Recording can also be done (probably easier) from the Audio button menu on the Menu Bar at the top of the screen. Actually, leaving an *Audio Record* line in your sequence is not recommended, but it aids in timing audio files with graphics.



After selecting Record, the field cursor jumps backward to the File field and you are asked to enter a filename. The next menu is a list of sound speeds in terms of Bits Per Second (BPS). Choose from 8kHz, 16kHz, and 32kHz. One of these choices must be made for the Play method to work. The number you select must be supported by your sound board. kHz stands for kilohertz, or thousandths of cycles per second. Select a speed. After the Next field, TEMPRA Media Author records your input sound source. Again, this is directly related to the sound board being used.

### ***PlayAsnc***

The list of appropriate files appears, again, relevant to the board type. Choose a sound file to Play. PlayAsnc audio events play simultaneously with other events. Do not try to start a new PlayAsnc before the current file has finished playing. If you use multiple audio files in your sequence, be sure to insert a Quiet line between them. When you play audio asynchronously with other events and the sound is “broken” or skips, set the Blast\_Buf in the AUTOEXEC.BAT for smoother audio. Setting the Blast\_Buf is explained in the Technical Information Appendix of this manual under “Environmental Variables.”

### ***Quiet***

Adding this line somewhere after a PlayAsnc line will instantly stop any audio playing. Also, selecting Quiet from the Audio menu on the Menu Bar will stop an audio file from playing while you are writing your sequences. If you use multiple audio files in your sequence, be sure to put a Quiet line between them so the audio events do not collide.

## Other Fields

Label, Next, and Wait are fields common to most lines. Apply them as necessary.

## Chapter Tutorial

This Chapter Tutorial is named AUDIO.SEQ in the sequence directory. Following is a copy of the Workform and the steps necessary to recreate the sequence. To learn how to do an Audio event only, just recreate Line 2. To play a sequence, select the Play Sequence button or [F6] key.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Audio	Conga	PlayAsnc				
	BColor		Vrt_In	99	Solid f80000		
	BColor		Vrt_Out	99	Solid 000000		
	Video	TOUCAN2	AaPlay	96 P	129bx 339by 1R		
	Audio		Quiet				
	Audio	Conga	PlayAsnc				
	BColor		Dissolve	99	Solid 51c		
	BColor		Spr_IN	99	Solid 51c		
	Audio		Quiet				

*Line 1 sets the stage for Buzz's show.*

### Recreating Line 1

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

*Line 2 starts the music for Buzz's performance.*

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **Audio**.
2. The **Audio** menu pops up. Select **PlayAsnc**.
3. The **PlayAsnc** menu pops up. Select **Conga.VOC** or **Conga.WAV**.
4. The **Wait** menu pops up. Select **Don't Wait**.
5. The **Next** menu pops up. Press **Select** to accept the default.
6. The music begins to play; the Line Counter advances to Line 3.

*Lines 3 and 4 are BColor events and are aimed to get your attention before Buzz makes his appearance.*

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Vrt\_In**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 4.

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Vrt\_Out**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 5.

*Line 5 brings Buzz to the screen using a Video, AaPlay event.*

### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. The **Method** menu pops up. Select **AaPlay** at the bottom of the menu.
3. The **Files** menu pops up. Select **TOUCAN2.FLI** from the menu.
4. The **SP/DP** menu pops up. Set the speed at **95**. Select **OK**.
5. The **Parameters** menu pops up. A ghost rectangle appears for you to position the viewing window on the screen. Move the rectangle to the center of the screen and press **Select**. At the **Number of Repeats** menu, type in **4** and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The file of Buzz spinning plays; the Line Counter advances to Line 6.

*Line 6 introduces you to the Quiet method. Be sure to use this Audio event when you use more than one audio in a sequence.*

### **Recreating Line 6**

1. Select the box under **Type**. From the pop-up menu, select **Audio**.
2. The **Method** menu pops up. Select **Quiet**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The music stops; the Line Counter advances to Line 7.

*The rest of the lines are repeats so we could show you how to use more than one Audio event in your sequences.*

### **Recreating Line 7**

1. Select the box under **Type**. From the pop-up menu, select **Audio**.
2. The **Audio** menu pops up. Select **PlayAsnc**.
3. The **PlayAsnc** menu pops up. Select **Conga.VOC** or **Conga.WAV**.
4. The **Wait** menu pops up. Select **Don't Wait**.
5. The **Next** menu pops up. Press **Select** to accept the default.
6. The music begins to play; the Line Counter advances to Line 8.

### ***Recreating Line 8***

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Dissolve**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 9.

### ***Recreating Line 9***

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Spr\_In**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 10.

### ***Recreating Line 10***

1. Select the box under **Type**. From the pop-up menu, select **Audio**.
2. The **Method** menu pops up. Select **Quiet**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The music stops; the Line Counter advances to Line 11.
6. Rewind the sequence to Line 1, press **F6** to see it play.



## Chapter 7

# A BColor Event

BColor events create background colors that wipe onto the screen at runtime and save on disk space. Your choice of colors ranges from 16 to 16.7 million, depending on your display card and the display mode.

### **Type**

This event wipes a background of a single color onto the screen during your sequence. When you are in the Type field, select BColor. Your selection goes into the Type field and the Wipes menu appears.

### **Method**

The Wipes menu appears as soon as the BColor event is selected. This menu lists the ways you bring your background color onto the screen. The wipes are, for the most part, self-explanatory and are similar to those used in the film and television industry. Select a wipe and the Event Controller goes to the Sp/Dp menu.

There is a demo file in the sequence directory named BCOLOR.SEQ that demonstrates the wipes available. There are four choices that should not be used in a BColor event. They are Sand\_UP, Sand\_DN, Drip\_UP, and Drip\_DN. These wipes are recommended for use with Screen or Sprite

events. There is a Drip\_DN wipe used in the VIDEO.SEQ found in the \TEMPRA\SEQ directory. They take more time than the other wipes and present an interesting effect.



Sharp Boy says:  
For an interesting effect,  
try using opposite wipes to  
bring in your background  
colors and screen events.  
Bring your background in  
with a Diag\_LR and then  
bring in a screen event  
using Diag\_UL.

Wipe Name	Wipe Description
Lft_Rt	Left to Right
Rt_Lft	Right to Left
Tp_Bot	Top to Bottom
Bot_Tp	Bottom to Top
Vrt_In	Vertical In to Center
Vrt_Out	Vertical Out from Center
Hzr_In	Horizontal In to Center
Hzr_Out	Horizontal Out from Center
Dissolve	Random fill
Box_In	A Box from screen frame In
Box_Out	A Box from screen center Out
Diag_UR	Diagonal to Upper Right corner
Diag_UL	Diagonal to Upper Left corner
Diag_LR	Diagonal to Lower Right corner
Diag_LL	Diagonal to Lower Left corner
Slide_U	Slide Up from the bottom
Slide_D	Slide Down from the top
Slide_L	Slide from the Left
Slide_R	Slide from the Right
Sand_UP	Image appears to dribble up the screen
Sand_DN	Image appears to dribble down the screen
Drip_UP	Image appears line-by-line
Drip_DN	Image appears line-by-line
Spr_In	Spiral In from frame
Spr_Out	Spiral Out from center
Preload	Load into memory for immediate display
Quick	Fastest fill
Rollup*	Roll up from bottom

\* Smooth Rollup in 320x200 256 and 640x480 16 color modes.

---

**NOTE:** *Preload sets up the screen buffer for later use. When you Preload a Screen or BColor, you load it into the replacement buffer for the visible screen. When you preload a screen before moving a Sprite, you reveal the new Screen in the trail of the Sprite. This happens because, as you move a Sprite across the visible screen, the part of the screen it travels over will be replaced with the new contents of the replacement buffer. You have to be careful to preload screens that are made in the same video mode as your current mode; otherwise, you will get unexpected results. You can preload for other effects as well. When you want to draw with a brush that is set to the background, you have to preload a BColor or Screen into the background (replacement) buffer. When you draw, you bring the background buffer into the visible screen with the drawing command. To preview this, you must use the [F6] previewing, not [F5].*

---

## Sp/Dp

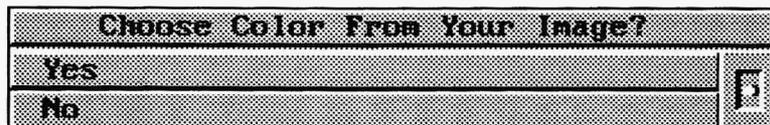
---

The Speed menu determines how fast your wipe is going to bring the background color onto the screen. The numbers, 1 to 99, represent the speed with 1 being the slowest and 99 the fastest. Select a speed. For more information on speed, see the Common Fields chapter in this manual.

## Parameters

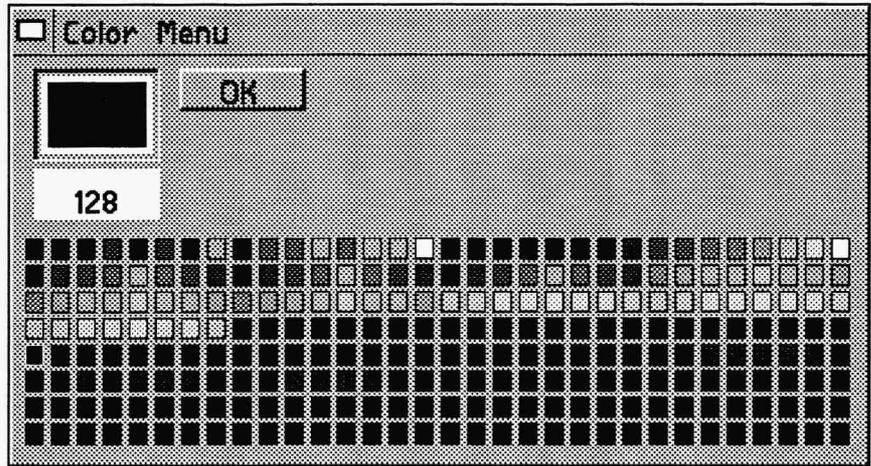
---

For this event, the Parameters field is where you select what color you want for your background color. You are asked if you want to choose the color from the screen or from a color palette.



If you select Yes, a small cursor with a colored box appears on your image. Move the cursor over the image until you find the color you want for your background color and press select. If you select No, a color palette appears. With the mouse, select the color you want. Then select OK. You have selected your background color. In 256-color mode, this Bcolor will change if you

bring in a graphic file that has a different palette than your current palette. The graphic file determines what colors are available in any given palette. You can set the color palette when you start building your sequence by inserting a Video event, LoadPal method. Instructions for the LoadPal event can be found in the Video chapter in this manual.



## ***Other Fields***

---

The Label, Wait, and Next are fields common to most lines. Apply them as necessary.

## ***Chapter Example***

---

This example wipes in a solid color diagonally, 50 pixels at a time starting at upper right corner.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
BColor	BColor	Diag_UR		50	Solid 34C		



## Chapter 8

# A Button Event

Use this event as a way of letting users interact with your sequence. This event allows for user input from a touch screen, mouse, keyboard, or joystick. Choices are predetermined in this branching event.

### **Type**

---

A Button event allows the user to move a Sprite across the screen as a pointer to button (select) areas on the screen. The system can identify the location of the cursor once the [Enter] key, left mouse button, or fire button on a joystick has been pressed. If the location of the cursor is within the boundary of a Button, then a branch occurs.

### **File**

---

Because the Button event allows the user to move Sprites as a screen pointer, this field is where the Sprite file to be used is selected. See the Sprite event chapter in this manual for information on creating sprites. (For touch screen users, there are no files to pick.)

### ***Default Cursor***

In the Button event, if you leave the File field blank, by pressing [Esc] while in the Sprite field menu, TEMPRA Media Author supplies an arrow for you to use. You may find it helpful to use this arrow cursor rather than supply a Sprite to act as a screen pointer.

### ***Method***

---

After the screen pointer is selected, the menu for the Method choices appears. Here you can pick up to 99 Button areas that you can make active with this Event.

### ***Parameters***

---

If you are editing an existing Button event, you will want to choose “Overwrite”; choose “Insert Lines” if you are creating a new Button event. The next thing you see is a message on your screen asking you to put the pointer at the beginning position. After all, the pointer has to show up some place to start off.

You have the option of putting an “R” after this position. This “R” makes the pointer’s position relative, so every time the user comes back to this event, the Sprite pointer is where he left it. Be sure you save the sequence after you have set the starting position where you want it, and before you run the event. The “R” modifies the position as the sequence runs.

It is helpful if you have the correct graphic file on the screen at this time. First you have to describe a safe active area. This keeps the user from flying the pointers off the perimeter of the monitor, or over a portion of the screen from which you want to keep them. Set the lower left limit and then the upper right limit to the invisible box that is the safe active area.

Now you have to delineate the active Button areas. This is done by positioning the pointer at the lower left corner of an imaginary box that is the active Button area and pressing [Enter] or Select, then moving the pointer to the upper right hand corner of the same imaginary box and pressing [Enter] or Select. When you have done this you can see the prompt for the Next field. Put in the Label

of the event you want to branch to when the pointer is found in this Button active area. Repeat this process for each active area.

In the process of defining a Button event, the field cursor pops between the Parameter and Next fields to set up a table. The table is as long as the number of items chosen in the Method field. This table contains the x,y coordinates for the active areas in the Parameters field and the Label names of the events in the Next field. Each of these event Labels are the lines you want to branch to when that Button is selected. Fill in the prompts and press [Enter] to set up the branch table. This is a process of popping between the Parameters field and the Next field until the Button is set up.

The first line of the Button event has the added feature of letting the user escape from a Button. If the user presses the [Esc] key (or cancel on mouse) while he is on a Button, the system automatically branches to the Label in the Next field on the first line of the event. If nothing is put in this Next field and [Esc] is pressed, control passes from the menu to the event with the Label that matches the first Next in the Button event.

## ***Next***

---

The Nexts are used to define where the TEMPRA Media Author system goes as a result of landing in an active area. The entry in the Next field on the same line as the Button coordinates tells TEMPRA Media Author which Label is the target of the branching. Enter the Label name of the event where you want to go as a result of the choice made.

Make sure all the labels in the Next fields match with Labels within the current sequence. The system tells you if this is true or not when you preview the sequence with [F6] or when you save the sequence. If you are using a touch screen, use the Configure option from the Workform Main Menu to set the touch screen option in the MA.CFG file. You, as the author, can toggle between touch screen and mouse use by pressing [Alt] T.

## Other Fields

Sp/Dp and Wait are fields common to most lines. Apply them as necessary.

## Chapter Tutorial

This Chapter Tutorial is named BUTTON.SEQ in the \SEQ directory. Following is a copy of the Workform and the steps necessary to recreate the sequence.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480		
	Video	TUTOR1	LoadPal				
	Global		Colors		47 167B 191 71T 15 4H		
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	99			
	Text	Q1	TIMES30	P 001	35x 412y 153c 24w 4h nnl		
	Sprite	HAND		15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		TIMES30	P 001	Rover #53x 174y 153c		
	Title		TIMES30	P 001	Buzz @238x 174y 153c		
	Title		TIMES30	P 001	Donatello @422x 174y 153c		
quest	Button	HAND	3 Areas	99 E	142bx 288by		quest
					-68bx -25by 574ex 457ey		
					-34bx 87by 119ex 163ey		wrong
					135bx 87by 288ex 163ey		right
					307bx 87by 565ex 163ey		wrong
wrong	Text	WRONG	TIMES30	E 001	170x 276y 157c 10w 4h nfm	3 0/18	quest
right	Text	RIGHT	TIMES30	E 001	136x 306y 130c 10w 6h nfm	3 0/18	

*Lines 1 - 5 set the stage to begin the Button event. When you are creating on your own computer, Line 1 may not be necessary. We have used Line 1 to be sure that everyone can run the sequence and learn about Button events.*

### **Recreating Line 1**

1. Select the box under **Type**. From the pop-up menu, scroll down and select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. The **Method** menu automatically pops up. Select **LoadPal**.
3. The **Files** menu pops up. You will select an image file on which to base the 256-color palette. Since the first image we use in the sequence is TUTOR1.PCX, that should be the palette to load. Select **TUTOR1.PCX**.
4. The **Wait** menu pops up. Select **Don't Wait**.
5. The **Next** menu pops up. Press **Select** to accept the default.
6. The Line Counter advances to Line 3.

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **Global**.
2. From the **Global** menu, select **Colors**.
3. A menu appears that allows you to set the colors for the borders of the boxes you will be creating later in this sequence. You can also select the text background and foreground colors. As you select from the menu, you are asked if you want to **Choose Color From Your Image?** Select **No**. Make your choices from the color menu. When you are finished select **Done**.
4. The **Wait** menu pops up. Select **Don't Wait**.
5. The **Next** menu pops up. Press **Select** to accept the default.
6. The Line Counter advances to Line 4.

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX**.
3. The **Method** menu pops up. From the **Wipe** menu, select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The background image appears; the Line Counter advances to Line 5.

### **Recreating Line 5**

1. Select the box under **Type**. Select **Screen**.
2. The **File** menu pops up. Select **TUTOR2.PCX**.
3. The **Method** menu pops up. From the **Wipe** menu, select **Bot\_Tp**.
4. The **SP/DP** menu pops up. Set the speed at **10**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Buzz wipes onto the caveman image; the Line Counter advances to Line 6.

*Line 6 is a Text event using text we created earlier and saved as a .TXT file in the \TEMPRA\TXT directory. This event line presents the question to be answered during the Button event.*

### **Recreating Line 6**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **Q1.TXT**.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use, or scroll down and select **Times 30**.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the first line in "What is the name of..." For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears on the screen. Move it to the color you want the text to be and press **Select**.
6. Framed text appears in the top left corner. Move the frame around the text

you want to display and press [Enter]. The text appears on the screen for positioning. Move the text toward the tip of Buzz's top blade (that's feather to you) so the text is visible even when the Menu Bar is on the screen and press **Select**.

7. The **Wait** menu pops up. Select **Don't Wait**.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 7.

*Line 7 is a Sprite event. The "hand" sprite will be the pointer used during the Button event that follows.*

### **Recreating Line 7**

1. Select the box under **Type**. Select **Sprite**.
2. The **File** menu pops up. Select **HAND.PCC**.
3. The **Method** menu pops up. Select **Simple** (for a basic sprite movement).
4. The **SP/DP** menu pops up. Set the speed at **15**. Select **Erase**. Select **OK**.
5. The **Parameters** menu pops up. Position the hand's starting point off-screen on the left side about eye level to Buzz, press **Select**. Then position the ending point near Buzz's eye, press **Select**.
6. The **Wait** menu pops up. Select **Wait for Timeout**. Type **2** on the **Seconds** menu. Press **Select**. Press **Select** again to accept the default of 0 on 18th Seconds.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The hand moves to the right, pointing at Buzz; the Line Counter advances to Line 8.

*Lines 8 - 10 are Title events. They place the three choices the viewer has to answer the question asked in Line 6.*

### **Recreating Line 8**

1. Select the box under **Type**. Select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use or scroll down and select **Times30**.
3. The **SP/DP** menu pops up. Select **Permanent** and select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **Rover**. Press **Select**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears on the screen. **Select** a color from the image.

5. Text appears for positioning. Move the text toward the bottom left of the screen (above where the Event Controller would be so you can see the name when the Event Controller is on the screen) and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 9.

### ***Recreating Line 9***

1. Select the box under **Type**. Select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use or scroll down and select **Times30**.
3. The **SP/DP** menu pops up. Select **Permanent** and select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **Buzz** and press **Select**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears. Position over a color you want from the image and press **Select**.
5. Text appears on the screen for positioning. Move the text toward the bottom middle of the screen and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 10.

### ***Recreating Line 10***

1. Select the box under **Type**. Select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use or scroll down and select **Times30**.
3. The **SP/DP** menu pops up. Select **Permanent** and then **OK**.
4. **Parameters** pops up. In the **Input Title** menu, type **Donatello** and press **Select**. For **Choose Color From Image**, select **Yes**. Using the cross-shaped cursor, select a color from the image.
5. Text appears on the screen for positioning. Move the text toward the bottom right of the screen and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 11.

*Lines 11 - 15 define the active area on the screen and set up the button areas around the three answer choices given in lines 8 - 10. These lines also designate the right and wrong answer in the Next fields.*

### **Recreating Line 11**

1. Select the box under **Label**. Type **quest** and press [Enter].
2. The **Type** menu pops up. Select **Button**.
3. The **File** menu pops up. Select **HAND.PCC** from the list.
4. The **Method** menu pops up. For **How Many Areas?**, type **3**, press [Enter].
5. The **SP/DP** menu pops up. Set the speed at **99** and **Erase**. Select **OK**.
6. The **Parameters** menu pops up. For **Insert Lines For Buttons**, select **Insert Lines**. The sprite appears with a message to position it at its starting point. Move the sprite to the tip of Buzz's beak and press **Select**.
7. The **Next** menu pops up. For **Next Label for [Esc] on Button?**, type **quest**, press [Enter]. (This tells TEMPRA ? to go back to the question if the viewer presses the [Esc] key.)
8. A message appears in the upper left corner of the screen. This is for Line 12.

### **Recreating Line 12**

1. The **Parameters** menu was already up while you were in Line 11. This contained a message to position the cursor at the lower left corner of the active area. Move the sprite cursor to the lower left corner of the screen and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the upper right corner of the screen and press **Select**.
3. Another message appears (the same as in the first message above). This is actually the beginning of Line 13.

### **Recreating Line 13**

1. The **Parameters** menu was already up while you were in Line 12. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where "Rover" is positioned. Move the cursor to the lower left corner of the word Rover and press **Select**.

2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the upper right corner of the word Rover and press **Select**.
3. The **Next** menu pops up. For **Area 1**, type in **wrong** as the label and press **[Enter]**. This causes a branch to the wrong label every time “Rover” is selected. At the wrong label during the sequence, text appears to indicate the wrong answer has been selected.
4. Another message appears (to determine another button area). This is the beginning of Line 14.

### ***Recreating Line 14***

1. The **Parameters** menu was already up while you were in Line 13. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where “Buzz” is positioned. Move the cursor to the lower left corner of the word Buzz and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the upper right corner of the word Buzz and press **Select**.
3. The **Next** menu pops up. For **Area 2**, type in **right** as the label and press **[Enter]**. This causes a branch to the right label every time “Buzz” is selected. At the right label during the sequence, text appears to indicate the right answer has been selected.
4. Another message appears (to determine another button area). This is the beginning of Line 15.

### ***Recreating Line 15***

1. The **Parameters** menu was already up while you were in Line 14. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where “Donatello” is positioned. Move the cursor to the lower left corner of the word Donatello and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the upper right corner of the word Donatello and press **Select**.

3. The **Next** menu pops up. For **Area 3**, type in **wrong** as the label and press **[Enter]**. This causes a branch to the wrong label every time “Donatello” is selected. At the wrong label during the sequence, text appears to indicate the wrong answer has been selected.
4. Press **Cancel** to return to the menus. You have completed the Button event for branching! The Line Counter advances to Line 16.

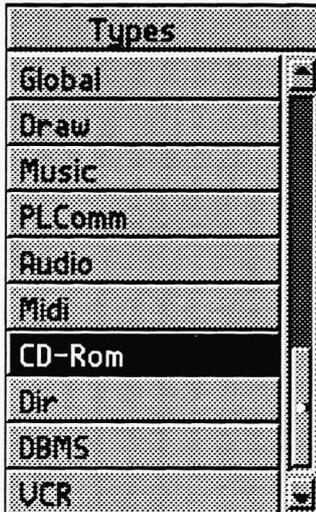
*Lines 16 and 17 are Text events that contain the “right” and “wrong” answer responses. Notice they have Labels to identify them.*

### ***Recreating Line 16***

1. Select the box under **Label**. Type in **wrong**. (Be sure to type the label exactly the same as the Next you used in Lines 13 and 15.)
2. The **Type** menu pops up. Select **Text**.
3. The **File** menu pops up. Select **WRONG.TXT**.
4. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
5. The **SP/DP** menu pops up. Select **Erase After Displaying**.
6. **Parameters** pops up. For **Choose Starting Line For Text**, select the first line of the words “Sorry, that’s wrong...” For **Allow User to Operate**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **Yes**. Select a color from the image using the cross-shaped cursor.
7. Framed text appears in the top left corner. Move the frame around the text you want to display (in this case, “Sorry, that's wrong. Please try again.”) and press **Select**. This generates a frame for the text. The frame appears on the screen for positioning. Move the frame to the **center** of the screen and press **Select**.
8. The **Wait** menu pops up. Select **Wait For Timeout**. In the **Seconds** box, type **3**. Press **Select**. Press **Select** again to accept 18th Seconds default.
9. The **Next** menu automatically pops up. Type **quest**. (This branches the sequence back to Line 11, which is labeled quest, so you can select the correct answer.)
10. Framed text appears on the screen; the Line Counter advances to Line 17.

### ***Recreating Line 17***

1. Select the box under **Label**. Type in **right**. (Be sure to type the label exactly the same as the next you used in Line 14.)
2. The **Type** menu pops up. Select **Text**.
3. The **File** menu pops up. Select **RIGHT.TXT**.
4. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
5. The **SP/DP** menu pops up. Select **Erase After Displaying**.
6. **Parameters** pops up. For **Choose Starting Line For Text**, select on the first line of the words “That’s right! Let’s continue.” For **Allow User to Operate**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **Yes**. Select a color from the image using the cross-shaped cursor.
7. Framed text appears in the top left corner. Move the frame around the text you want to display (in this case, “That’s right! Let’s continue.”) and press **Select**. This generates a frame for the text. The frame appears on the screen for positioning. Move the frame to the **center** of the screen and press **Select**.
8. The **Wait** menu pops up. Select **Wait For Timeout**. In the **Seconds** box, type **3**. Press **Select**. Press **Select** again to accept 18th Seconds default.
9. The **Next** menu pops up. Press **Select** to accept the default.
10. Framed text appears on the screen; the Line Counter advances to Line 18.
11. Rewind the sequence to Line 1 and press F6 to see how it works.



## Chapter 9

# A CD-ROM Event

Add high quality audio to your sequence. This event allows you to play and control access to audio tracks recorded on a CD-ROM.

### **Type**

Select the Type field and the event menu pops up. Select CD-ROM and the Method menu pops up.

### **Method**

You can access audio on CD-ROM various ways.

### **Play**

This method allows you to play an audio bite or music track from a CD-ROM during a sequence. The playing commences where the play head is positioned. You should use the Search method first to position the play head over the section of a particular track you want in your sequence.

### **Playrang**

This method allows you to play a range of audio tracks from a CD-ROM during a sequence. CD-ROMs are designed so there are tracks which contain audio, but each track is divided into smaller,

precisely accessible units. This makes for very tight editing of sound and music bites. To get to a place within a track, you have to locate it in terms of minutes, seconds, and frames (75 frames per second). A CD-ROM may contain many tracks. Each track starts at zero minutes:zero seconds:zero frames, 00:00:00.

### ***Seek***

You should position the play head before you choose Play or Playrang. This is a good idea because some CD-ROM players are not very fast at finding programmed tracks. By pre-positioning the play head, you can get the audio to start with the Play method exactly when you want it to. Remember, you can designate exactly where you want the play head to be positioned. To do this, tell TEMPRA Media Author what track number you want and how far into that track you want to go.

### ***Pause***

You can suspend the playback of audio from the CD-ROM. Once the Pause method is executed, you can restart the audio playback with Play or Continue. (Use the one that works with your CD-ROM player. They are not all the same.)

### ***Continue***

Once the playback of the CD-ROM audio has been paused, you can restart the playback with Continue. The CD-ROM Method allows you to continue playing your preselected audio track from the CD-ROM.

### ***Stop***

Use Stop to make the playback of the CD-ROM audio stop. After you use this method, you need to use Play to restart the playback of audio. Most often you would perform a Stop, then a Seek, then a Play. You might also execute a Playrang.

### ***Load***

Some CD-ROM players support a computer-controlled Load command. If yours does, this method gets the disk up and running.

## ***Eject***

Use this method to eject the CD-ROM. Naturally, this causes the playback of audio to stop, and you can change disks at this point. Eject is the opposite of Load and when used together automates changing of CD-ROMs.

## ***Parameters***

---

Parameters are only necessary when you use the Playrang and Seek methods.

### ***Playrang***

When you are using the Playrang method of the CD-ROM event, you have to designate the range you want to play. Select the track number, then the place on the track (i.e., seconds or frames) you want to start from. These places are described in terms of the time it takes to get to them from the start of the track. For example, you can start 2 minutes, 30 seconds into the track. If you need to choose a place that is between the second markers, select a frame. There are 75 frames to a second. Since you are playing a range, you also have to set the end point for the audio to stop. This end point is also described in terms of tracks, minutes, seconds, and frames.

### ***Seek***

When you use the Seek method of the CD-ROM event, you have to designate the place on the CD-ROM where you want to position the play head. Select the track number, then the place on the track you wish to be. These places are described in terms of the time it takes to get to them from the start of the track. For example, you can start 2 minutes, 30 seconds into the track. If you need to choose a place that is between the second markers, select a frame. Once the play head is positioned, you can use the Play method and the audio will play from exactly where you have positioned the play head with the Seek method.

## ***Other Fields***

---

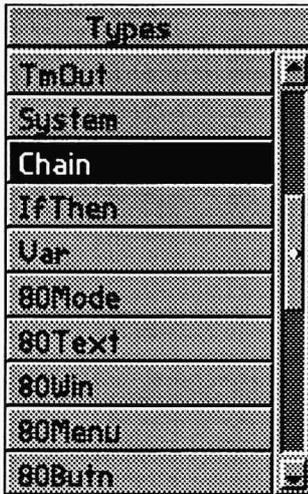
Label, Wait, and Next are fields common to most lines. Apply them as necessary.

## ***Chapter Example***

---

This example causes the CD-ROM to begin playing.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	CD-Rom		Play				



## Chapter 10

# A Chain Event

Don't be confined by the limits of computer memory. With the Chain event you can link sequences together to create mega sequences that are as long as you like. Each individual sequence can be no more than 500 lines long.

### *Type*

The Chain command is placed at the point in the sequence where you want to start a different sequence. When the Chain command is executed, the next event you see is the first event of the chained sequence. Select the box under the Type field, the event menu pops up. Select Chain. Chain appears in the Type field.

Once the chained sequence begins to play, control passes to it. When you chain to another sequence from the middle of a sequence, the rest of the first sequence does not play. You can continue to chain together as many sequences as you want by placing the Chain event in each sequence in the order you want them to run. In other words, the Chain event from sequence 1 to sequence 2 would be placed where you want to begin the chain in sequence 1. If you then want to chain from sequence 2 to sequence 3, the Chain event would be in sequence 2 where you want to begin the Chain to sequence 3 and so on.

## ***File***

---

When you select Chain from the Type menu, a directory menu appears. This is a menu of sequences. Choose the sequence you want to chain to from your current sequence, and press [Enter]. Your choice is entered onto the screen and the field cursor goes directly to the Wait field.

## ***Wait***

---

Don't Wait, Wait for Any Key, Wait for Timeout pertains to the wait that you want before the chain to the new sequence is executed. This is different from the Wait For in other events where the waiting occurred after the event. In the Chain event, execution of the Chain line takes program control away from the previous sequence and a new sequence is now in memory. Normally, you would choose Don't Wait to go directly on to the new sequence.

When a new sequence is loaded, TEMPRA Media Author loads the paths associated with that sequence, if the paths have been saved with the same name as the sequence name. Using Chain will allow you to change paths as you run sequences.

## ***Other Fields***

---

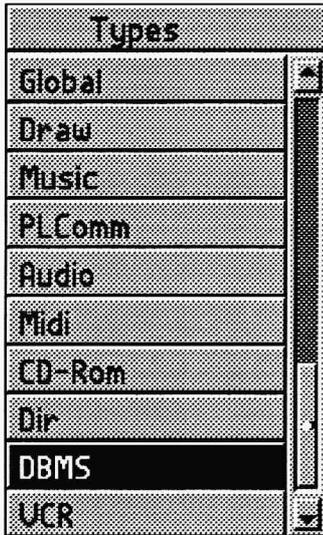
Label and Next are fields common to most lines. Apply them as necessary. Method, SP/DP and Parameters do not apply to the Chain event.

## ***Chapter Example***

---

This example shows a Chain event that starts a sequence named "Key."

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
Chain	Chain	KEY					



## *Chapter 11*

# ***A DBMS Event***

Connect your databases with multimedia using the DBMS event. This event allows you to use a database in conjunction with your multimedia application. You can also make a database with TEMPRA Media Author. You can read, write, and add to an existing database file (DBF) and use the information in the fields in your presentation.

Users of this event should be familiar with database programming. The DBMS event is included to provide compatibility with other dBase III compatible DBMS systems.

### ***Type*** \_\_\_\_\_

To create or use a database with TEMPRA Media Author, select DBMS from the Type menu. This brings up the File menu to select how you want to use the database.

### ***File*** \_\_\_\_\_

The File menu gives you a list of choices for using or creating a database.

### ***Index***

Choose from the NDX files listed in the directory menu. The NDX files reside in one of two directories: either in the directory you set for the DBMS files using the PATH menu; or, if you

did not set any directory, these files reside in the same directory as your TEMPRA Media Author program. On execution, the chosen index becomes the current one.

### ***MakeDBF***

The file field holds the name of the Data Base File (DBF) that you want to create.

### ***MakeNDX***

The file field holds the name of the index file that you want to create.

### ***Memo2txt***

The file field holds the name of the \*.DBT file that you want to extract the memo from. This \*.DBT file is a database memo file that has the same prefix as the DBF file that it is associated with.

### ***Open***

Choose the name of the DBF file from the Directory menu. This becomes the open database file upon which all database actions occur. **Only one database can be open at a time.** You can open one, use it, close it; then open a second one, all within one sequence.

### ***ReIndex***

The file field holds the name of the index file that you want to reindex using the currently open DBF. Remember that TEMPRA Media Author allows only one database file to be open at a time.

### ***Txt2memo***

The file field holds the name of the \*.DBT that you want to add the text to. This DBT file has the same prefix as the DBF file that it is associated with.

### ***Method***

---

The Method menu pops up for you to select how you want to use the files you've selected for the database.

Database
Append
Close
Delete
FieldNam
Goto
Index
IndexOff
Lock
MakeDBF
MakeIDX

## ***Append***

Append adds a blank record to the end of the DBF file. The blank record then becomes the current record.

### **Append Modifies:**

\$\_dmax (number of records in a DBF file)

\$\_drec (number of the current record)

\$\_ibot (number of the last record)

## ***Close***

Used to close the current database file. **Only one database can be opened at a time.**

### **Close Modifies:**

\$\_dbf (variable holding the name of the current DBF)

## ***Delete***

Delete flags the current record as deleted. The record is not actually removed from the DBF until the DBF is packed. To check the deleted flag for the current record, test the \$\_ddel variable. \$\_ddel contains the string "DELETED" if the record is deleted.

### **Delete Modifies:**

\$\_ddel (variable holding "DELETED" if record is deleted)

## ***FieldNam***

The Field Name method is used to extract the actual name of a specific field from a record in a DBF file and place that field name in a variable.

## ***Goto***

Goto goes to an absolute record (1 is the first record).

### **Goto Modifies:**

\$\_drec (number of the current record)

## ***Index***

Sets the current index to the index file specified by the file field. Index files are independent of the DBF files they reference, so you must be careful to use the right index file for the DBF of your choosing. The DBF is viewed as being sorted by the index, even though the records have not actually been moved. See **Skip**.

TEMPRA Media Author supports a circular array of five active indexes. The most recent Index command designates the current index.

### **Index Modifies:**

`$_ndx` (name of the current index)

`$_itop` (number of the first record using the active index)

`$_ibot` (number of the last record using the active index)

## ***IndexOff***

Turns off all index files.

### **IndexOff Modifies:**

`$_ndx` (the name of the active index)

`$_itop` (the first record using active index)

`$_ibot` (the last record using active index)

## ***Lock***

Locks the current record. This command is used in network environments to prevent two or more users from updating the same record at the same time. When TEMPRA Media Author is used on a network, the environmental variable `VV_SHARE` should be set to allow records to be shared by multiple users.

## ***MakeDBF***

This method is used to create a DBF file. MakeDBF uses an ASCII text file as a specification file with the suffix `SPC`. The text file is used to describe the structure of the database. You specify the field names, types, width, and precision (precision only used with numbers).

**MakeDBF Modifies:**

`$_dbf` (the name of the active DBF)

***MakeNDX***

This method is used to create an index file. See the Parameters description for more information.

**MakeNDX Modifies:**

`$_ndx` (the name of the active index)

***Memo2txt***

This method is used to extract a memo field from the current record and place the memo in an ASCII text file. See the Parameters section description for more information.

***Open***

This method is used to open a DBF. Before you can use a DBF you must open it. Be sure to open a DBF only once, and to close it before you open any other. Only one DBF can be used at a time.

**Open Modifies:**

`$_dbf` (the name of the active DBF)

`$_dmax` (the total number of records in the DBF)

`$_drec` (the current record)

`$_itop` (the first record using active index)

`$_ibot` (the last record using active index)

***Pack***

Pack is used to physically remove all records from the currently open DBF that have been flagged as deleted. Pack also updates active indexes to reflect the changes.

**Pack Modifies:**

`$_d-del` (DELETED flag if the record is so marked)

`$_dmax` (the total number of records in the DBF)

`$_drec` (the current record)

`$_itop` (the first record using active index)

`$_ibot` (the last record using active index)

### ***Read***

Read is used to extract the current record from the DBF and create/modify variables so that they contain the contents of the fields from the current record. The variables are made in series with a common first three characters. For example. `$f1000`, `$f1001`, `$f1002`.

### ***ReadFld***

ReadFld is used to extract a single field from the current DBF record and place the contents in a variable.

### ***ReIndex***

Reindex updates the index file specified in the file field so that it is up to date with respect to the current DBF file. It also makes the named index the current index. Reindex is generally used to bring a nonactive index file up-to-date.

#### **ReIndex Modifies:**

`$_dmax` (the total number of records in the DBF)

`$_ndx` (the name of the current active index)

`$_itop` (the first record using active index)

`$_ibot` (the last record using active index)

### ***Rlock***

Rlock locks the current record and is intended for use in a network environment.

### ***Seek***

Seek is used to find a record that matches the key specified in the Parameters field. Seek searches the current index for a matching key. If the key is found, Seek positions the DBF record pointer to the found record.

#### **Seek Modifies:**

`$_drec` (the current record number)

`$_key` (the key that is found as a result of the search)

## ***Skip***

Skip moves the record pointer relative to the current position. Skip uses the current index if there is one.

### **Skip Modifies:**

`$_drec` (the current record number)

### **For example:**

You have an open DBF file with a list of names and addresses in no particular order. Your current index is up-to-date and is keyed on the zip codes. Your current record points to a person with a zip code in the 19004 postal area. A skip of 1 would position the current = 19004. A skip of -1 would position the current record pointer at the previous record with a zip code less than 19004.

## ***Txt2memo***

This method is used to replace the memo field of the current record with the contents of an ASCII text file. See the Parameters description for more information.

## ***Write***

The Write method is used to update all of the fields in the current record with the contents of the variables described in the parameters field. This requires the use of series variables only in number sequence.

## ***WriteFld***

The WriteFld method is used to update a single field of the current record.

## ***Undelete***

The Undelete method flags the current record as undeleted. See also Delete.

### **Undelete Modifies:**

`$_ddel` (removes DELETED flag)

### **Unlock**

The Unlock method unlocks the current DBF file and all records in the file. This allows other users on a network to update it.

### **Parameters**

---

The Parameters field designates the variables with which to access and control the database.

### **FieldNam Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		FieldNam		\$field < #num		
	DBMS		FieldNam		\$field < 2		
	DBMS		FieldNam		\$field < \$num		

#### **Where:**

“\$field” is the variable that will receive the field name.

“#num” is an integer variable holding the field number.

“2” is the field number.

“\$num” is a variable holding the field number.

### **Goto Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Goto		\$num		
	DBMS		Goto		5		

#### **Where:**

“\$num” is a variable holding the record number you want to go to.

“5” is the record number you want to go to.

#### **Goto Modifies:**

\$\_drec (current record number)

## MakeDBF Example

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		MakeDBF		\$spec		
	DBMS		MakeDBF		spec.spc		

### Where:

“\$spec” is a variable holding the DBF specification file name.

“spec.spc” is the name of the specification file. The file must reside in the DBMS directory to be found by TEMPRA Media Author.

A specfile contains a description of each field in the DBF. The syntax is: **FIELDNAME,FIELD\_TYPE,LENGTH,DECIMALPLACES.**

- The field name must be less than 11 characters in length and may **not** contain any spaces.
- The maximum record length is 4000 bytes. This number is the total of all lengths of all the fields in the record.
- The maximum number of fields is 128.
- The maximum length of a character field is 254. Character fields cannot be longer than 100 characters if it is used as a key.
- The maximum length of a numeric field is 19. The number of decimal places must be less than 16 and greater than or equal to zero and also less than length -2.

The field types are as follows:

**C** - character

**N** - number and precision # of decimal points

**D** - date MM/DD/YY

**M** - memo

**L** - Logical (0 or 1, length of 1)

**Example format:**

Field	NameField	Type	Length	Decimal Places
	a_fname,	C,	20,	0
	a_lname,	C,	23,	0
	a_money,	N,	10,	2

**Always use these values for memo, date, and logic fields**

a_memo,	M,	10,	0
a_logic,	L,	1,	0
a_date,	D,	8,	0

The actual text file would include the following lines BETWEEN the == == marks.

```

===== this line is not a part of the specfile
a_fname,C,20,0
a_lname,C,23,0
a_money,N,10,2
a_memo,M,10,2
a_logic,L,1,0
a_date,D,8,0
===== this line is not a part of the specfile
    
```

***MakeNDX Example***

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS	Fname	MakeNDX		\$keyex Character		
	DBMS	Fname	MakeNDX		\$keyex Numeric		
	DBMS	Fname	MakeNDX		\$keyex Character Unique		
	DBMS	Fname	MakeNDX		\$keyex Numeric Unique		

**Where:**

“\$keyex” is a variable holding the key expression. The key expression is the name of the field as it appears in the SPC file, and may be used in combination with the functions listed below.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Define		\$keyex - "a_fname"		
	DBMS	fname	MakeNDX		\$keyex Character		

Creates an index filename. The index orders the records in the current DBF.

**Character** creates an index using a character-based key expression.

**Numeric** creates an index using a numeric field.

**Unique** creates an index with unique keys for each record.

The key expression for character based key expressions can only contain one or more field names and may be combined with the following functions.

**DTOC(field\_name)** converts data in a date field to a date string DD/MM/YY.

**LEFT(field\_name,#)** returns the left-most characters in the field. The numbers of characters returned is set by #.

**RIGHT(field\_name,#)** numbers of characters returned is set by #, counting from right from end of field.

**SRT(field\_name,length,decimal\_places)** converts from numeric to character field expresses numbers as numerals.

**SUBSRT(field\_name,start,count)** returns count characters from the middle of the string, starting at position start.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$keyex +- "RIGHT (DTC(date))"		
	DBMS	fname	MakeNDX		\$keyex		

Creates an index on the year MM i.e. 12/19/56 would be 56.

### **Memo2txt Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		FieldNam		\$field < #num		
	DBMS		FieldNam		\$field < 2		
	DBMS		FieldNam		\$field < \$num		

**Where:**

“\$tmp” is a variable that holds the filename for the extracted memo.

“tmp.txt” is the filename for the extracted memo.

“4” is the field number of the memo field in this example.

“a\_memo” is the field name of the memo field.

### **Read Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Read		\$f1		

**Where:**

“\$f1” is the var prefix.

Var \$f1001 will hold the contents of field 1.

Var \$f1002 will hold the contents of field 2, etc.

These variables will be created when the event is executed.

### **ReadFld Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		ReadFld		\$field < NAME		
	DBMS		ReadFld		\$field < \$name		
	DBMS		ReadFld		\$field < 6		

**Where:**

“\$field” is the variable that will receive the contents of the field.

“NAME” is the field name.

“\$name” is a variable holding the field name.

“6” is the number of the field that is being read.

**Seek Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Seek		\$key		
	DBMS		Seek		\$key FIRST		

**Where:**

“\$key” is a variable holding the key to search for.

The optional parameter **first** flags the first logical record that matches the key.

**Seek Modifies:**

\$\_drec and \$\_key

**Skip Example**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Skip		\$num		
	DBMS		Skip		3		

**Where:**

“\$num” is a variable that holds the number of records to skip.

“3” is the number of records to skip, in this example.

**Skip Modifies:**

\$\_drec

### ***Txt2memo Example***

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Txt2Memo		\$tmp > a_memo		
	DBMS		Txt2Memo		tmp.txt > a_memo		
	DBMS		Txt2Memo		\$tmp > 2		
	DBMS		Txt2Memo		tmp.txt > 2		

**Where:**

“\$tmp” is a variable that holds the filename for the text that replaces the memo.

“a\_memo” is the field name of the memo field.

“tmp.txt” is the filename for the text that will replace the memo.

“2” is the field number of the memo field in this example.

### ***Write Example***

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		Write		\$f1		

**Where:**

“\$f1” is the var prefix.

Var\$f1001 will be written to field 1.

Var\$f1002 will be written to field 2, etc.

If var \$f1### does not exist, the field(s) will be blank.

### ***WriteFld Example***

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS		WriteFld		\$data > 4		
	DBMS		WriteFld		\$data > name		
	DBMS		WriteFld		\$data > \$field		

**Where:**

“\$data” is a var containing the data to be written.

“4” is the field number.

“name” is the field name.

“\$field” is a var containing the field name.

***Memo2txt Example***

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	DBMS	DB_FILE	Memo2Txt				

TEMPRA Media Author looks for \*.DBT file in the file field.

## **DATABASE EXAMPLES**

<p>Specification file:  <b>IMAGES.SPC</b>          name,C,8,0          ext,C,3,0          memo,c,35,0</p>	<p>The data file:  <b>IMAGES.DAT</b>          SLIDE20,PCX,This is the Tempra Box.          TUTOR2,PCX,This is Buzz in paradise.          TOUCAN2,FLI,This is Buzz in flight.          TUTOR1,PCX,This is Buzz's paradise.</p>
---	---

**Example one:**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$sep = ",",		
	Var		Declare		\$infl = "images.dat"		
	Var		Declare		\$inln = " "		
	Var		Declare		#cntr = 1		
	Var		Declare		\$key = "NAME"		
	DBMS	IMAGES	MakeDBF		images.spc		
	DBMS	IMAGES	Open				
read	Var		Read		\$infl #cntr \$inln		
	IfThen				\$inln = "^Z"		exit
	Var		Modify		#cntr += 1		
	Var		Split		\$inln \$sep \$fld 1		
	DBMS		Append				
	DBMS		Goto		\$_dmax		
	DBMS		Write		"fld*" 1		read
exit	DBMS		Close				

This example shows a specification file, a text file, and how the text file is converted to a DBF file. First, declare the variables you need to read in the text file. "\$infl" holds the name of the text file we want to read in.

“\$inln” holds the whole line of text as it is read in . “\$sep” holds the separator character we use to split apart the incoming text line. The Var Split event creates the variables “\$fld001, \$fld002, and \$fld003” as a result of the Split. “#cntr” is a numeric Var used to advance as we read successive lines in the text file. (See the chapter on Vars for more details on how variables work.)

After the declarations, the example starts reading in each line, splitting the elements apart and putting each of these elements into the database file named IMAGES. The DBMS Append EVENT makes a new record. Goto “\$\_dmax” positions you at the new record. Write "fld\*" 1 writes all fields to the new record, starting with "fld001".

### Example two:

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$temp = "TUTOR1"		
	DBMS	IMAGES	Open				
	DBMS		Goto		\$_itop		
read1	DBMS		Read		\$fl 1		
	IfThen				\$fl001 >= \$temp		show
	IfThen				\$_drec == \$_dmax		exit
	DBMS		Skip		1		read1
show	Screen	\$fl001	Quick			any key	
exit	DBMS		Close				

The second example assumes you have done example one and that the screens listed in the first field of the DAT file are in your screen directory. If you have installed TEMPRA Media Author on your hard drive, the necessary files will be present.

The example above opens a DBF called "IMAGES." Starting at the first record, the DBF is read. Field 001 is tested to see if the contents match a known item, "\$temp." If there is no match, then the example shows how you proceed to the next record with a Skip and loop back to the reading and checking. If there is a match, the sequence branches to the event line that has the label SHOW. Then the contents of field 001 replaces a var label used to hold the filename for a Screen event. The Screen event will display the file identified with the variable "\$temp", which was declared at the top of the sequence to be "TUTOR1".

Please notice that we check for the end of the DBF by comparing the record number of the current record "\$\_drec" with the number of the last record stored in "\$\_dmax."

### ***Internal Database Variables***

**All internal database variables are string variables.**

- \$\_dbf** database file name if one is open.
- \$\_ddel** string "deleted" (in caps) if record is deleted, delete flag.
- \$\_dmax** number of the last record in file.
- \$\_drec** number of current record.
- \$\_key** key that was found from a Seek.
- \$\_ndx** file name of the active index
- \$\_itop** record number of the first record listed in the index. If no index is used, then \$\_itop is the first record in the DBF.
- \$\_ibot** record number of the last record listed in the index. If no index is used, then \$\_ibot is the last record in the DBF.

### ***Other Fields***

---

Label is a field common to most lines. Apply as necessary.



## *Chapter 12*

# *A Dir Event*

When you make commercial applications with TEMPRA Media Author your users will need to access files through the Dir event.

### ***Type*** \_\_\_\_\_

The Dir event is used to give the user a directory from which to choose a file. The Dir event gives you the flexibility to get any kind of file from any directory on the drive. What you do is create a string variable with the directory path and a mask for the file type. The Dir event uses this to bring up a list of files. When the user chooses a file, the name of that file goes into a string variable of your choosing. Then you can use that string variable with the filename as a variable in the file field of a subsequent event.

### ***File*** \_\_\_\_\_

You can use the default variable `$_file` to hold the filename. You can also enter a string variable of your own that you have already declared. If you use the `$_file` variable you may find that subsequent Dir events modify the contents of this variable. If you want to make sure the filename in the string variable does not change, you should use your own variable from the start.

## Method

---

You are asked to “Input Var with Path and Mask.” This is where you enter the name of the variable you have that holds the path to the files and the mask for the file type.

*NOTE: You should have already created a string variable with the DOS path to your file type. For example, if your text files are in a directory called “TEXT” you can make a string variable “\$TEMP” and set it equal to “\TEXT.” Next you have to make sure the mask is correct for your file type. Without a mask, all the file types are shown in the directory on the screen. With a mask you can choose to show files with only one suffix. A mask “\*.TXT” brings up only files ending with TXT. To add this mask to the path, modify your string variable “\$TEMP” with the “+=” modifier and “\\*.TXT.”*

---

In the Parameters field, you should have \$TEMP += “\\*.TXT.” The easiest way of getting the path to the various kinds of files is to use the reserved variable that holds the path information from the \*.PTH file. You can have this in a Var event to set up this variable: \$TEMP = &Text, then in the next line: \$TEMP += “\\*.TXT” sets the mask. If you use this method the path becomes sequence specific because you are loading the Path from the \*.PTH file. As you remember the \*.PTH file is loaded with each new sequence.

Once you have the right values in this variable you put the name of the variable into the Method field.

## Sp/Dp

---

Speed doesn’t matter, but you have to choose a display type.

**Permanent** - sticks all foreground events to the background.

**Temporary** - stays in the foreground until further notice.

**Erase After Display** - removes all foreground events.

## Parameters

First you are asked to title the directory. Enter a title that explains what kinds of files you are going to show in the directory. Next you are asked to position the directory on the screen. Use the mouse or cursor keys to do this and press [Enter] when the directory is where you want it. The directory will be displayed in the default font set with the Global font event.

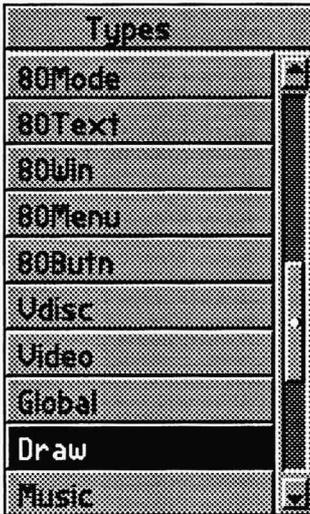
## Other Fields

Label, Wait, and Next are fields common to most lines. Apply as necessary.

## Chapter Example

This example sets up the Dir command with Vars, and then creates a directory titled "Demo Dir". The chosen file name is stored in the var `$_file`.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$mask = ""		
	Var		Modify		\$mask = &program		
	Var		Modify		\$mask += "\*.exe"		
DirY	Dir	\$_file	\$mask		31x 19y "Demo Dir"		



## Chapter 13

# A Draw Event

Draw events let you create basic shapes, like circles or squares, either filled or unfilled. You can use the shapes to contain text, graphics, or other image files. You can use the Draw event to create your own wipes using preloaded images or BColors, or with freehand you can annotate your graphics.

### ***Type***

---

After selecting or pressing [Enter] on the Type field, select Draw from the Type menu. The Draw event lets you create your own wipes with a background brush, make colored squares, lines and more. This is very useful in saving disk space. It also lets you make very powerful and exciting visual presentations.

### ***Method***

---

This field is used to determine which of six different types of drawing tools you want to use.

### ***Line***

This draws a straight line between two points.



Sharp Boy says:  
Use a BColor event first, selecting white as your background color...it's easier to draw on a white screen. You can always delete the BColor line when you're finished drawing, if you want a black background.

### ***Circle***

This draws an empty ellipse.

### ***FCircle***

This draws a filled ellipse.

### ***Box***

This draws an empty rectangle using two corner points.

### ***FBox***

This draws a filled rectangle using two corner points.

### ***Set Brush***

This sets the size, shape, and color of the drawing tool.

### ***Freehand***

This lets you draw freehand designs on the screen. Freehand allows a speaker to annotate an image during a presentation.

## ***Parameters***

---

The Parameters you choose are determined by what drawing method you are using.

### ***Line***

You set the beginning position and ending position of the line by selecting two points on the screen with the mouse or cursor keys (to move the screen cursor) when prompted.

### ***Circle***

You set the center position by selecting on the screen with the mouse, or cursor keys when prompted. You size the Circle with the mouse or cursor keys, as well. As you move the mouse, the ellipse is drawn on the screen so you can see what you are doing.

### ***FCircle***

You set the center position just like sizing and positioning an empty circle by following the prompts and using the mouse, or cursor keys, to move the screen cursor. As you move the mouse, the ellipse is drawn on the screen so you can see what you are doing.

### ***Box***

You set the lower left position and upper right position when prompted on the screen, using the mouse or cursor keys to move the screen cursor.

### ***FBox***

You set the lower left position and upper right position just like sizing and positioning an empty box by following the prompts and using the mouse, or cursor keys, to move the screen cursor.

### ***Set Brush***

Use this to set the shape, color, and drawing attributes of the brush you are using. You are first asked to set the brush shape (either a circle or a square). The next menu that appears is the attribute menu. Of five possible choices listed, AND, OR, and XOR are logic terms and have specific meanings in computer science. (When using these features, experiment. Experimentation is the best way to understand the effects these functions have on the objects drawn.)The other two are Background and Set. Background draws whatever you have in the background buffer. Set allows you to choose the size of the drawing tool, and then the color. This is the color all of your lines and shapes are until you change it using Set Brush.

### ***Other Fields***

---

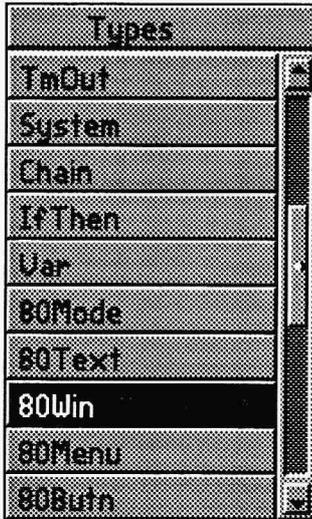
Label, Next, and Wait are fields common to most lines. For more information, please refer to the Common Events chapter.

## ***Chapter Example***

---

This example sets the drawing brush to a 7-pixel circle. The second event draws a line.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Draw	SetBrush		T	Circle Set 7s 39c		
	Draw	Line		T	36bx 49by 236ex 181ey		



## Chapter 14

# 80Butn Event

80Butn events are the same as Button events, except in an 80 column format. The tutorial in the Button event chapter of this manual will help you set up an 80Butn Event. The only difference is that the Button event is set up in graphics mode and the 80Butn event is set up in 80 column text mode.

### **Type**

---

A Button event allows the viewer to move a Text pointer across the text screen and use it as a pointer to square areas, 80Butns, on the screen. The system can identify the location of the Text pointer once the [Enter] key, left mouse button, or fire button on a joystick has been pressed. If the location of the Text pointer is within the boundary of an 80Butn, a branch occurs.

### **Method**

---

Use the Method field to choose how many button areas you are going to designate.

## ***Sp/Dp***

---

Choose the maximum speed at which you want the pointer to travel while you are moving the pointer with the cursor keys.

## ***Speed***

You use a joystick or the cursor keys to move the pointer and keep the pressure on the device, the pointer accelerates to the maximum speed indicated in the Sp/Dp field.

## ***Display***

You have the choice of making these Text pointers Permanent, Temporary, or Erasing after the Key press. This is a design decision. If you want pointers to build up on the screen, use Permanent. If you want to erase the pointer and other temporary events, choose Erase after Displaying.

## ***Parameters***

---

If you are editing an 80Butn event, you probably want to choose “Overwrite”; otherwise choose “Insert Lines.” The next thing you see is a text window on your text screen asking you to put the Text pointer at the beginning position. After all, the Text pointer has to show up some place to start off. You have the option of putting an “R” after this position. This “R” makes the Text pointer position relative, so every time you come back to this event, the Text pointer is where you left it, not where it started off. Save your sequence after you set the starting position where you want it and before you run the event. The “R” modifies the position as the sequence runs.

Tell the system what Label event you want to branch to if the user presses the [Esc] key or right mouse button during this event, by entering that Label in the first Next.

You have to describe a Safe Active area. This keeps the user from flying the Text pointers into outer space or over a portion of the screen you want to keep them from. Set the lower left limit and then the upper right limit to the invisible

box that is the Safe Active area. Now you have to delineate the active 80Butn areas. This is done by positioning the Text pointer at the lower left corner of an imaginary box that is the active 80Butn area and pressing [Enter] or the left mouse button. Next move the Text pointer to the upper right hand corner of the same imaginary box and press [Enter] or the left mouse button.

After you have done this, you see the requester window for the Next field. Put in the Label of the event you want to branch to when the text pointer is found in this active area. Repeat this process for each active area.

In the process of defining an 80Butn event, the field cursor pops between the Parameters field and the Next field to set up a table. The table is as long as the number of items chosen in the Method field. This table contains the x,y coordinates for the 80Butns in the Parameters field, and the Label names of events in the Next field. These are the events you want to branch to as a result of selecting an 80Butn. Fill in the requester windows with the Labels of these events and press [Enter] to set up the branch table. This is a process of popping between the Parameters field and the Next field until the 80Butn is created.

## ***Next***

---

The Next field contains the Label name that represents the target event for the menu choice on the same line.

## ***Other Fields***

---

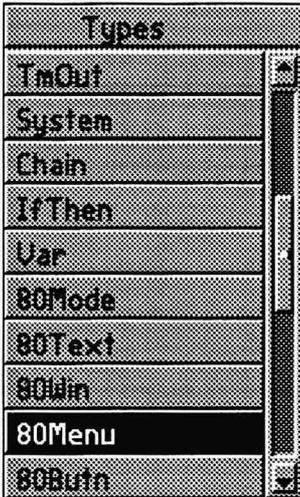
Label and Wait are fields common to most lines. Apply them as necessary.

## Chapter Example

---

This example creates two button areas on the 80 column text screen and allows branching to events “one” and “two.” Pressing the [Esc] key takes you to a line marked “escap.”

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	80Butn		2 Areas	p	38bx 16by		escap
					0bx 0by 78ex 23ey		
					2bx 2by 18ex 7ey		one
					27bx 11by 54ex 14ey		two



## *Chapter 15*

# *80Menu Event*

This event puts a “pop up” menu on the text screen and gives the user/viewer the ability to pursue one of the avenues represented by the item he chooses from the list. Up to 99 choices are possible. Choices can be made with the cursor keys on the keyboard, a mouse, or a joystick.

The tutorial in the Menu event chapter of this manual will help you set up an 80Menu event. The only difference is that the Menu event is set up in graphics mode and the 80Menu event is set up in 80 column text mode.

### ***Type***

When the field cursor is in the event Type field, and the Event Menu pops up, highlight “80Menu” and press [Enter]. Your choice appears in the type field.

### ***Method***

The field cursor jumps from the Type field to the Method field. You see a menu offering 2 to 10 items. Choose a number based on the number of choices you want to give the user/viewer. If you want to give him more than 10 choices, select 10 from the How Many Items? menu and continue setting up the menu. To edit the number of choices, move the cursor over the Method box, press

F3 and type in the number you want. You can also go to the Workform by selecting the Work button on the Menu Bar. Go to the Menu line in the method column. Press F3 and type in the number. Now you can go back and add the additional choices. If you have already positioned and sized the box, you may need to resize and reposition the box to include the additional choices. Remember, the user can scroll through the menu if they do not all fit on the screen in the box you have made.

### ***Sp/Dp***

---

You have the choice of making these menus Permanent, Temporary, or Erasing after the selection. This is your design decision. If you want menus to build up on the screen, then use Permanent. Otherwise, use Temporary or Erasing.

### ***Parameters***

---

First tell the system whether you want to insert new lines into the form or overwrite the current lines. If you are editing an 80Menu, you probably want to choose overwrite; otherwise, choose "Insert Lines."

The next thing you see is a prompt asking for the title of the menu. Notice every menu in the TEMPRA Media Author system has a title. This is your place to tell your user/viewer what your menu is about.

Fill in the requester windows and press [Enter] to set up the branch table. This is a process of popping between the Parameters field and the Next field until the Menu is set up.

The table is as long as the number of items in your 80Menu as chosen in the Method field. This table contains the list of choices you are giving the user/viewer in the Parameters field, and the Label names of the events you want to branch to in the Next field.

Now a menu is on the text screen. Use the mouse or cursor keys to size the window around the Menu. Then use the mouse or cursor keys to position that menu where you want it to be seen. When this is done press [Enter].

## ***Editing 80Menu with Prompts***

You can edit the 80Menu Event by positioning the field cursor in the Parameters field and pressing [Enter]. You see a menu that allows you to change one aspect of the menu, or all.

### ***All***

You can recreate the menu by choosing to edit all the aspects of the menu.

### ***Title***

You can change only the title of the 80Menu by selecting this.

### ***Menu Items***

You can change only the items listed in the 80Menu by selecting this.

### ***Width/Height***

You can change the size of the menu by selecting this.

### ***Position***

You can change the placement of the menu on the Text screen by selecting this. To change the color of the menu, use the Global event, Colors method.

## ***Next***

---

The Next fields are used to define where the TEMPRA Media Author control goes as a result of choosing one of the items on the menu list. The Next on the same line as the menu choice shows which event line will be executed once the menu is terminated. Enter the Label name of the line you want to play as a result of leaving the menu. The special line of the Menu event with the title allows for the added feature of letting the viewer escape from a menu.

## ***Other Fields***

---

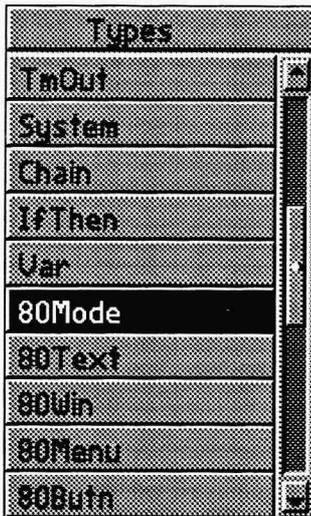
Label and Wait For are fields common to most lines. Apply them as necessary. For more information, please refer to the chapter on Common Fields.

## Chapter Example

---

This example makes a menu in 80 column mode with two choices: “Item One” and “Item Two.”

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	80Menu		2 items	p	35x 8y 10w 4h		
					Demo Menu		escap
					Item One		one
					Item Two		two



## Chapter 16

# 80Mode Event

This Event is used to switch between graphic color mode and 80 column color text mode.

### **Type**

Select 80Mode from the Type menu by selecting on the Type field.

### **Method**

The Method field lets you turn the 80 column color text mode On, Off, or Clear.

#### **On**

Use this to turn the 80 column color text mode on.

#### **Off**

If you determine you want to turn on the VGA Graphics screen display, then turn this OFF. The graphic screen will look just the same as it did when you left it. To switch to graphic mode and have a new Screen or BColor, simply select one of those events.

### **Clear**

Use Clear if you want to clear the 80 column text screen.

### **Other Fields**

---

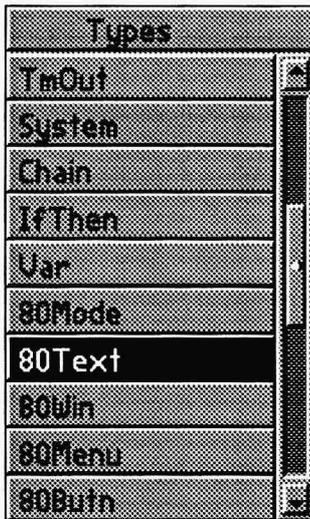
Label, Next, SP/DP, and Wait are fields common to most lines. For more information on these fields, please refer to the Common Fields chapter in this manual. Parameters and File do not apply.

### **Chapter Example**

---

This example switches the output of the video screen to 80 column text.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	80Mode				On		



## Chapter 17

# 80Text Event

80Text events are the same as regular Text events, except in the 80 column format. No font selection is available because 80Text uses the PC ROM font.

### **Type**

---

Use this event to put ASCII text files on the screen with the color attributes defined in the most recent Global event, Colors method.

### **File**

---

After you select the 80Text type, you see a Directory menu. Use this to select the file you want to display.

### **Sp/Dp**

---

After choosing a Display type, choose which line of the text file you wish to start displaying. The text file comes onto the screen. Scroll through the file (using the cursor keys or mouse) until you select the line you want to be the first line showing on your display screen. When the selection is made, the line number is put into the Workform.

### ***Parameters***

The Parameters field is used to determine the color of the text and whether you want a frame around it. If you want a frame, you are given the opportunity to allow the user to operate the window. Then you size and position the window. When you are creating the 80Text event for the first time, you are given all these choices in order. When you are editing, you are presented with a menu asking which of the options you want to modify.

### ***Position***

Use your cursor keys or mouse to position your text. Then press [Enter]. The text coordinates appear in the Parameters field. 80Text events have only one set of coordinates. Press [Ctrl] X to see the x,y coordinates in the corner of the screen.

### ***Color***

This is where you choose the color of the characters (foreground color). If you choose to pick the color from the color palette, select the color you want from the colors in the small squares.

### ***Width/Height***

Once the operation question is answered for framed windows, you set the size of the window. The text from the file is shown on the screen. Use the cursor keys or mouse to stretch the window to fit the text.

### ***Not Framed***

If you want the text to be written over the text screen without a frame, answer No to the “Framed?” question.

### ***Framed***

If you want the text in a framed window, answer Yes to the “Framed?” question. A window frame has the color attributes set with the global values defined with the last Global event. The Global values for the windows at startup are the same ones used in the Workform. To change the color attributes, select Global from the Type menu, then Colors from the Global menu. A menu pops up that allows you to change the default colors.

## ***Operate/Frame***

If you choose Framed text, you can allow the user to operate the text window by scrolling through it with the cursor keys. It also allows the user to pan right and left, using the arrow keys, to see long lines of text that do not fit into the text window. A window that cannot be operated only displays the text you choose.

## ***Justify***

You have the option of justifying the text on the left or right of the window, or to center the text in the window.

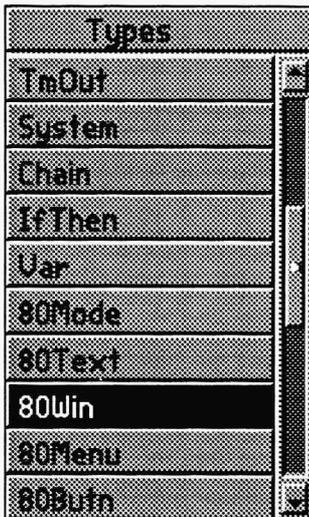
## ***Other Fields***

Label, Next, and Wait are fields common to most lines. Please refer to the Common Fields chapter of this manual.

## ***Chapter Example***

This example puts file DEMO.TXT on the 80 column text screen, starting with Line 3.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	80Text	DEMO		T 003	26x 12y 8c 33w 5h nn		



## Chapter 18

# 80Win Event

80Win events are much like sprites, except they use text characters rather than graphics to be moved around on the screen. This event type is useful in displaying text windows made with the PWIN Utility. To run PWIN, select Subprogram from the Workform Main Menu. See the PWIN Utility Appendix in this manual for help in using PWIN. The files you create will have a .WND extension. These windows can be made to look like any PC text screen in 80 column mode, so you can reproduce data entry screens for training mock-ups and so on.

### **Type** \_\_\_\_\_

Select 80Win from the Type menu.

### **File** \_\_\_\_\_

After you select the 80Win type you are given a directory menu to the right of the cursor. Use this to select the file you want to display.

## ***Sp/Dp***

---

Display is not used, but you will need to choose a Speed.

## ***Speed***

This determines how fast the WND moves across the screen.

## ***Parameters***

---

Use the cursor keys or mouse to position the .WND file on the screen. When it is where you want it, press Select or [Enter]. You can choose a second x,y pair if you want to move the text WND file across the screen. If you do not want it to move, press [Enter] again and both x,y pairs are the same and the .WND file does not move. You can see two x,y coordinate pairs in the Parameters field.

## ***Other Fields***

---

Label, Next, and Wait are fields common to most lines. Method does not apply. For more information, see the chapter on Common Fields in this manual.

## ***Chapter Example***

---

This example puts DEMO.WND file on the 80 column color text screen.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	80win	DEMP		1 T	12bx 15by 12ex 15ey		



## Chapter 19

# A Global Event

Global events set the conditions for the entire sequence and reset previous Global events. You can override the global settings in some events. For instance, you can have a Global event set the default font to one of the many KST fonts. You can override the default font temporarily with a font choice in a Text event without changing the default font.

### **Type**

---

Select Global from the Type menu.

### **File**

---

The only Global setting that requires a file is the Font setting. Once this is set to the font you want, all Text events and Menu events are shown in this font, unless otherwise specified.

## ***Method***

---

You can change various values globally in the Method field.

### ***ResetAll***

This resets all global values to their defaults.

### ***Colors***

This determines all Text display colors: Text, Border, and Highlight. You see the current setting for text that is active in TEMPRA Media Author while you choose your colors. Highlight colors are used in menus, user-operated text windows and input variables.

### ***Font***

This determines the font that is used to show all menus, directories, ticker, scrollup, title and text, unless a new font is picked in the Text Event or Title Event. The current Global font becomes the default font.

### ***Dropshad***

You can set a drop shadow to appear with all the text. Choose the x and y offset and color.

### ***DropOff***

This turns the drop shadow off.

### ***Autoset***

TEMPRA Media Author automatically adjusts the screen to the right resolution and palette for the chosen image.

### ***AutoOff***

You can turn off the automatic detection of screen resolution and palette loading with this selection.

### ***Delay***

This determines a delay to slow down wipes, sprite movement, or drawing. This is a handy feature for today's super-fast PCs.

**ScaleOn**

Set this to ensure that all screen images are automatically scaled to the visible screen if they have the right number of colors.

**ScaleOff**

Set this to turn automatic scaling off.

**QuickOn**

Set this to maximize speed of full screen display when you do not have sufficient memory in your computer for the mode you are using. When this is set, TEMPRA Media Author ignores all wipe methods and brings your screens to the visible screen as fast as possible. If EMS is detected, then Quick On is ignored.

**QuickOff**

Set this to turn off the above Global command.

**BackOn**

Set this to let your viewer back up through the sequence while its running using the [F9] key. This also allows them to skip ahead with the [F10] key. When TEMPRA Media Author starts, the default for this feature is "on." It must be turned off if you want to disable it.

**BackOff**

Set this to turn off the BackOn feature.

**Style2D**

Set this to get a two dimensional appearance to the text frames, menus and directories.

**Style3D**

Set this to get a three dimensional appearance to the text frames, menus and directories. This is the default when you start TEMPRA Media Author.

**StyleTL**

Set this style to get the look of the TEMPRA Media Author GUI in text frames, menus and directories. The font color is always black when using this style.

### ***PIC\_PCX***

Set this to tell TEMPRA Media Author to look for 8-bit PCX files in the Screen directory.

### ***PIC\_CEG***

Set this to tell TEMPRA Media Author to look for CEG files in the Screen directory.

### ***PIC\_TGA***

Set this to tell TEMPRA Media Author to look for 16- and 24-bit (16.7 million color) TGA files in the Screen directory.

### ***SPR\_PCC***

Set this to tell TEMPRA Media Author to look for 8-bit PCC files in the Sprite directory. This is automatically set when you are in 16- or 256-color modes.

### ***SPR\_CLP***

Set this to tell TEMPRA Media Author to look for CLP files in the Sprite directory.

### ***SPR\_WIN***

Set this to tell TEMPRA Media Author to look for 16- and 24-bit (16.7 million color) WIN files in the Sprite directory. This is automatically set when you are in high color modes.

## ***Parameters***

---

You can see in the Parameters field the color, delay, and drop shadow you are setting globally.

### ***Colors***

Setting the global color parameters allows you to set the colors for:

***Border Foreground***

This sets the foreground color for a framed box.

***Border Background***

This sets the background color for a framed box.

***Text Foreground***

This setting controls the color of the text characters.

***Text Background***

This setting controls the color of the area behind the text characters.

***Highlight Foreground***

This sets the highlighted text color in a framed box the user can operate.

***Highlight Background***

This sets the highlight background color in a framed box the user can operate.

You can choose the color from the graphic screen or a color menu. If you choose to get the color from the screen, you are presented with a cursor that has a square border. The cursor's border color is the selected color. Move the cursor over the color you want to select and press [Enter]. If you choose to pick the color from the color palette, select the color you want from the colors in the small squares.

***Delay***

You can set the delay to be from 0 to 999 computer cycles. We have included this to allow you to slow the Draw, Wipe, and Sprite speeds if you have a very fast computer.

***Dropshad***

You choose the offset of the drop shadow in the x and y direction, then using the color menu, or the image itself, you select the color for the drop shadow.

## ***Other Fields***

---

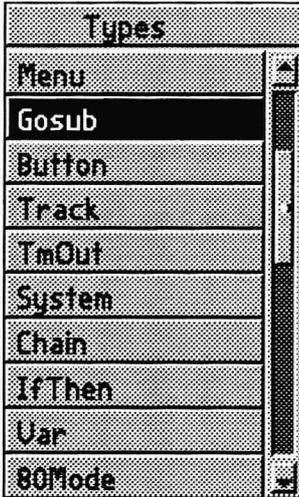
Label, Next, SP/DP, and Wait are fields common to most lines. Apply them as necessary.

## ***Chapter Example***

---

This example shows a Global event that sets the Font to Times20.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Global	Times20	Font				



## Chapter 20

# A Gosub Event

Subroutines produce efficient applications. As a multimedia creator, you often find that you want to do the exact same thing a number of times, at a number of different places in your presentation. With the Gosub event, you can do this and still have only one copy of the event lines that do the repetitive work. All you have to do is “call” this series of lines to work from wherever you want the work done. Since you can call subroutines in remote sequences, you can use the work you have done before in your current project, or even have a sequence of only subroutines that are called from other sequences.

### **Type**

This event type is used to branch to a specially created set of event lines, which automatically return control to the line following this event when finished. These events can be in the current sequence or in other sequences on the disk.

**NOTE:** *To set up a subroutine, all you need to do is give an event line a label, then put <ret> in the Next field of the last line of the subroutine. The return statement (<ret>) returns control to the line that follows the calling Gosub line.*

## **File**

---

If you want to get to a subroutine in the current sequence, select “**Subroutine in This Sequence**” (the top choice on the menu).

If the subroutine is in another sequence, select “**Subroutine in Another Sequence.**” You must then choose the sequence name for the file column. Do this by selecting from the list of files that appear on the screen. If the sequence has a different path, you can still get to it, but you have to enter the sequence name manually and that sequence must have a custom path file in the same directory with the same name as the sequence.

Don’t forget that once you have loaded a new path, you have to reload the old path to get back to the files you started with. To do this, make sure you have saved a path file with the same name as the sequences you want to call.

## **Next**

---

The Next column is filled in from a requester window and has to have the name of the label for the subroutine you are going to. Make sure the spelling is the same and has the same capitalization scheme.

---

***NOTE:** Make sure that a subroutine is entered only via a Gosub. Otherwise, when the <ret> is found by the system it looks for the line number of the non-existent calling routine and stays there. Pressing [Esc] returns control to you.*

---

## **Other Fields**

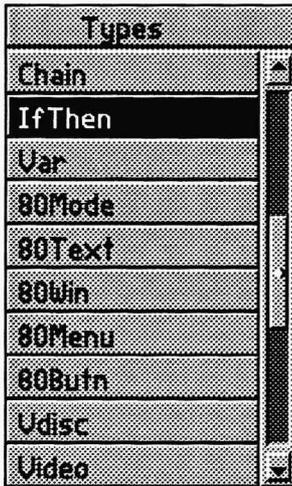
---

The Label, Method, Sp/Dp, Parameters, and Wait fields are common to most lines. Apply them as necessary.

## ***Chapter Example***

This example executes a subroutine labeled “sub1.” After the subroutine “sub1” has executed, the line following the Gosub line is executed.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Gosub						sub1



## Chapter 21

# *An IfThen Event*

Create internal branching the user doesn't control. The IfThen event is used to judge the existence of files, value of variables, as well as choices taken and stored in the answer track.

### *Type*

Select this type when you want to make a branch based on the value of a variable or an element of the viewer's answer track. If the result of the test described in the parameters field is true, then control passes to the label that matches the one in the Next field. If the result is not true, control passes to the following line. Event form : **IF** | parameter field test is **True** | **THEN** goto the label in next field, **Else** fall through to the following line.

### *Parameters*

The four kinds of values tested with the IfThen event are: **Answer** (from the answer track), **Numeric**, **String** (both are kinds of variables), and if a **File** exists. Choose the type of value you are going to test for: answers, the values of numeric variables, the contents of string variables, or the existence of a file.

### **Answers**

Answers are stored if you have “**Track On,**” and if the track has not been saved. The most recent choices made from menus and user windows can be found if the menus and user window events have been labeled. These answers are stored by their position in the list of possible answers. So choosing the second selection from a menu named “**M-1**” would be stored internally as “**M-1 = 2.**” Input the label for the menu or user window event.

#### **Choose the kind of test**

- ==** Equals
- !=** Does not equal
- >** greater than
- >=** greater than or equal to
- <** less than
- <=** Less than or equal to

Now you can test the number of the choice against a variable’s value, or against a number between 0 and 99 that you enter.

### **Variables**

Variables must be declared and given initial values. Then they can be used as parts of the IfThen event when they are called by name.

Numeric variables are different from string variables, so the tests are different.

Internal system variables starting with “&” come from the computer itself and are available for your use at any time. Month is “&month,” day of month is “&mday,” day of week is “&wday,” year is “&year,” hour is “&hour,” minute is “&min,” and second is “&sec.”

There are also internal system variables for the DBMS functions, the current directory path, and graphic mode.

## ***Numeric Variable***

Numeric variables are integer numbers and can be tested like numbers. In TEMPRA Media Author, all numeric variable names start with a “#.” Enter the name of the variable when asked to input a Var name.

### **Choose the kind of test**

==	Equals
!=	Does not equal
>	greater than
>=	greater than or equal to
<	less than
<=	Less than or equal to

You choose if you want to test the first variable against a different variable or a number you enter. Enter the Var name of the numeric variable, or the value of the number when requested. The variable you test against can be any system variable or one you defined.

## ***String Variable***

The contents of a String variable is made up of a list of characters. In TEMPRA Media Author, all string variable names start with a “\$”. A word is a string of characters, and so is a sentence. A space is a character, and so are numerals (numbers are not the same as numerals). When you test Strings you can test the contents of the list. You can also test the value of a string of numerals according to the value of that string as a real number.

Input the name of the string variable when the requester window asks for the Var name.

### **Choose the kind of test**

= =	The strings match exactly.
!=	The strings do not match exactly.
>	The test string is contained within the known.
>=	The test string is contained within the known but if they are equal that would be OK too.
<	The known string is contained within the test.

- <=     The known string is contained within the test but if they are equal that would be OK too.
- N==    Test strings as numbers to see if they match exactly.
- N!=    Test strings as numbers to see if they do not match.
- N>     Test strings as numbers to see if one is greater than a known.
- N<     Test strings as numbers to see if one is less than a known.
- N>=    Test strings as numbers to see if one is greater than or equal to a known.
- N<=    Test strings as numbers to see if one is less than or equal to a known.

Choose if the known is a different string variable or a literal string. For a different string variable, enter the Var label in the requester window. For a literal string, enter the string in the requester. Notice that there are only 15 spaces for your literal string.

If you choose to test for the existence of a file, simply enter the name of the file between quotes in the Parameters field or enter the string variable name that you are using to hold the name of this file.

### ***Next***

---

If your test proves to be true, then the system branches to the event Label that matches the contents of the Next field, otherwise (else) the control falls through to the following event.

### ***Other Fields***

---

Label, File, Method, Sp/Dp, and Wait are fields common to most lines. Apply them as necessary.

---

## Chapter Example

---

This example tests the values in the two Vars, “\$ans1” and “\$test”. If the values are equal, TEMPRA Media Author branches to the line labeled “this.”

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	IfThen				%ans1 == \$test		this



## Chapter 22

# A Menu Event

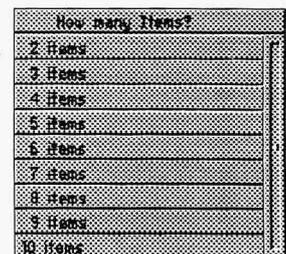
Allow your audience to interact with your multimedia application by making choices from menus. This event can be used with the mouse or touch screen to produce interactive presentations or kiosks.

### Type

Select the Menu event to place a pop-up menu on the graphic screen. From the Types menu, select Menu. You create a list of choices from which the viewer chooses and branches to other parts of your sequence. The Menu can contain as many as 99 choices and the user can scroll through the choices using the slide bar or arrow keys. Choices are made with the [Enter] key, the left mouse button, or fire button on the joystick.

### Method

The field cursor bypasses the File field to the Method field. You see a menu offering 2 to 10 items. Choose a number based on the number of choices you want to give the viewer. If you want more than 10 items, go ahead and select 10 from the How Many Items? menu and continue setting up the menu. To edit the number of choices, move the field cursor over the method box, press [F3], and type in the number of choices you want. You can also edit the number of choices by going to the Workform



by selecting the Work button. Select the Menu line in the method column of the Workform, press [F3] and type in the number. Now you can continue to set up the additional choices in your menu. To resize the box, select on the Parameters field, then select Width/Height and resize the box. Remember, if all of the choices aren't showing, the viewer can scroll through them. You may need to reposition the box, since it will be larger when you add items.

## Parameters

---

The Parameters field is where you set up your menu, giving it a Title, the number of items, and positioning it on the screen.



Sharp Boy says:  
Menu events let your audience get involved. Let your viewers make as many choices as possible. They'll love it.

### Create New Menus

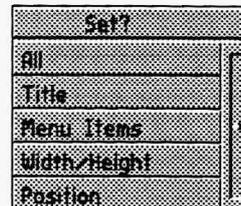
The first choice in the Parameters field is the insertion of new lines or overwriting the current lines in the Workform. If you are editing a Menu, you probably want to choose "Overwrite"; otherwise, choose "Insert Lines." The next thing you see is a prompt asking for the title of the menu. You may notice every menu in the TEMPRA Media Author system has a title telling you what the Menu is about.

Now you are prompted to set up branching by typing in the Label and Next for each menu item. This is a process of popping between the Parameters field and the Next field until the Menu is set up.

The table is as long as the number of items chosen in the Method field. This table is set up so that each time a menu option is selected by the user, TEMPRA Media Author takes them to the matching line in the sequence for a response to their choice. Now a menu appears on the screen. Use the mouse or cursor keys to size the window around the Menu. Then use the mouse or cursor keys to position the menu where you want it. Press Select or [Enter].

## ***Editing Menus with Prompts***

You can edit the Menu event by positioning the field cursor in the Parameters field and pressing [Enter]. You see a menu that allows you to change one aspect of the menu or all.



### ***All***

You can recreate the menu by choosing to edit all the aspects of the menu.

### ***Title***

This choice allows you to change the title of the created Menu.

### ***Menu Items***

You can change only the items listed in the Menu by selecting this. You can also edit directly onto the Workform to change these entries. See the Workform chapter in this manual for further explanation.

### ***Width/Height***

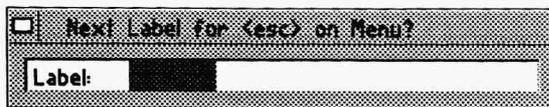
You can change the size of the menu by selecting this. The border of the menu is moved one character at a time. The Menu text appears in whatever font you have set in the last Global event, as explained in the Global event chapter of this manual.

### ***Position***

You can change the placement of the menu on the screen by selecting this. To change the color of the menu, select Global event from the Type menu, then Color from the Global menu.

## ***Next***

The Next fields are used to define where the TEMPRA Media Author control goes as a result of choosing one of the items on the menu list. The Next on the same line as the menu choices shows which event line is the target of the branching. Enter the La-



Label for <esc> on Menu?

bel name of the line where you want to go as a result of the choice made. The special line of the Menu event with the title allows for the added feature of letting the viewer escape from a menu. If the viewer presses the [Esc] key or right mouse button while he is on a Menu, the system automatically branches to the Label in the Next field across from the Menu Title. If nothing is put in this Next field, control passes from the menu and the event line directly after the Menu event is executed.

### ***Other Fields***

---

Label, File, and Wait are fields common to most lines. Apply them as necessary.

### ***Chapter Tutorial***

---

This Chapter Tutorial is named MENU.SEQ in the sequence directory. Following is a copy of the Workform and the steps necessary to recreate the sequence.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 256		
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
quest	Menu		3 items	P	176x 184y 33w 8h		
					WHAT IS MY NAME?		end
					Rover		wrong
					Buzz		right
					Donatello		wrong
wrong	Music		Play		t255 o2 c2 <c2		
	Text	WRONG	TIMES30	E 001	164x 279y 55c 10w 4h nfm	3 0/18	quest
right	Music		Play		t255 o4 c8 d8 e8 f8 g		
	Text	RIGHT	TIMES30	E 002	151x 270y 55c 11w 4h nfm	3 0/18	
end	BColor		Box_Out	99	Solid 151c		

---

*Lines 1 - 3 set the stage for the Menu event with Screen events to bring Buzz into view.*

### ***Recreating Line 1***

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

### ***Recreating Line 2***

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX**.
3. The **Method** menu pops up. From the **Wipe** menu, select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. **Parameters** pops up. At the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default. The **Next** field is used when branching to a **Label** name. The default is a blank **Next** field.
8. The image appears on the screen; the Line Counter advances to Line 3.

### ***Recreating Line 3***

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR2.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Bot\_Tp**.
4. The **SP/DP** menu pops up. Set the speed at **10**. Select **OK**.
5. **Parameters** pops up. At the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. You want to continue to the next screen automatically, so select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Buzz appears on the caveman background screen; the Line Counter advances to Line 4.

*This is a multiple line event that will automatically take you through the next five lines. The Label and Next names must match. TEMPRA Media Author distinguishes between capital letters and lower case letters, so be sure you type them the same or the sequence will not work.*

### **Recreating Lines 4 through 8**

#### **Line 4**

1. Select the box under **Label**. Type **quest**, press [Enter].
2. From the **Types** menu, select **Menu**.
3. In the **Method** field, a “**How Many Items**” menu appears. Select **3** items.
4. The **Sp/DP** menu pops up. Select **Permanent**.
5. In the **Parameters** field, the **Insert Lines** menu pops up. Select **Insert Lines**.

#### **Line 5**

6. The **Input Menu Title** box appears on the screen. Type in **What is my name?**
7. The **Next** field pops up on the screen. Type **end**.

#### **Line 6**

8. A box appears asking you for the first item. Type in **Rover**.
9. The **Next** field pops up; type **wrong**.

#### **Line 7**

10. A box appears asking you for the second item. Type in **Buzz**.
11. The **Next** field pops up; type **right**.

#### **Line 8**

12. A box appears asking you for the third item. Type in **Donatello**.
13. The **Next** field pops up; type **wrong**.
14. A box will appear in the upper left corner of the screen. Position the box in the lower half of the screen. Press [Enter]. The Line Counter advances to Line 9.

*Line 9 shows you how to create a few music notes to liven up your presentations. This is the “wrong” label event line. Appendix D, Creating Music, in this manual will help you create your own tunes. Don’t forget to check it out.*

## **Recreating Line 9**

1. Select the box under **Label**. You are going to assign a label to match the **Next** created in Lines 6 and 8. Lines 6 and 8 branch to Line 9. Type **wrong**. (Remember, the spelling and case must match the Next fields in Lines 6 and 8.) Press [Enter].
2. The **Type Menu** pops up. Scroll down and select **Music**.
3. The **Method** menu pops up. Select **Play**. You are going to create a few notes of music on the keyboard that will play during this branch.
4. The **Parameters** menu pops up. This is where you type the music notes to play through the computer’s speakers. Type **t255 o2 c2 < c2..** in the menu and press **Select**.
5. The **Wait** menu pops up. Select **Don’t Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The “wrong” answer music plays; the Line Counter advances to Line 10.

*Line 10 is a Text event that imports text from a file we created earlier and saved as a .TXT file in the \TEMPRA\TXT directory.*

## **Recreating Line 10**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **WRONG.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you want to use or scroll down and select **Times 30**.
4. The **SP/DP** menu pops up. Select **Erase After Displaying**.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select the first line of the words “Sorry, that’s wrong...”. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **Yes**. Select a color from the image using the cross-shaped cursor.
6. Framed text appears in the top left corner. **Move the frame around the text you want to display** and press **Select**. This generates a frame for the

text. The frame appears on the screen for positioning. Move the frame to the upper center of the screen and press **Select**.

7. The **Wait** menu pops up. **Select Wait for Timeout**. In the **Seconds** box, type **3**. Press **Select**. Press **Select** again to accept 18th Seconds default.
8. The **Next** menu pops up. Type **quest**. (This branches the sequence back to Line 4, which is labeled quest.)
9. Framed text appears on the screen; the Line Counter advances to Line 11.

*Line 11 is another creating music event with the “right” label.*

### **Recreating Line 11**

1. Select the box under **Label**. Type **right** in the menu. Press [Enter] to accept.
2. The **Type** menu pops up. Scroll down and select **Music**.
3. The **Method** menu pops up. Select **Play**.
4. The **Parameters** menu pops up. Type **t255 o4 c8 d8 e8 f8 g** in the menu and press **Select** (these are musical notes to play during the “right” answer event).
5. The **Wait** menu pops up. Select **Don’t Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The “right” answer music plays; the Line Counter advances to Line 12.

*Line 12 is another Text event with the “right” answer text..*

### **Recreating Line 12**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **RIGHT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you want to use or scroll down and select **Times 30**.
4. The **SP/DP** menu pops up. Select **Erase After Displaying**. This prompts the text to disappear after it is displayed rather than waiting for the next event to wipe it off the screen.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select on the first line in “**That’s right! Let’s continue..**” For **Allow User To Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select

- Middle.** For **Choose Color From Image**, select **Yes**. Using the cross-shaped cursor, select a color from the image.
6. Framed text appears in the top left corner of the screen. Move the frame around the text you want to display and press **Select**. Move the text to the center of the screen and press **Select**.
  7. The **Wait** menu pops up. Select **Wait for Timeout**. In the **Seconds** box, type **3**. Press **Select** again to accept the 18th Seconds default.
  8. The **Next** menu pops up. Press **Select** to accept the default.
  9. Text appears in the box; the Line Counter advances to Line 13.

*Line 13 is a BColor event to close out the presentation.*

### ***Recreating Line 13***

1. Select the box under **Label**. Type **end**.
2. The **Type** menu pops up. Select **BColor**.
3. The **Method** menu pops up. From the **Wipe** menu, select a wipe you want to try or select **Box\_Out**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select your choice of color and then select **OK**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The screen changes color and the sequence is finished.
9. Rewind the sequence to Line 1 and press **[F6]** to see how it works.



## *Chapter 23*

# *A Midi Event*

Use the Midi event to play music without storing large WAV files on your hard drive.

### ***Type***

Select or press [Enter] over the Type field and a menu appears. Select Midi. In order to play midi files, you will need an appropriate driver for your sound board. For more information on midi, see the Technical Information Appendix of this manual.

### ***File***

When you select Play from the Midi menu, the field cursor goes to the File field.

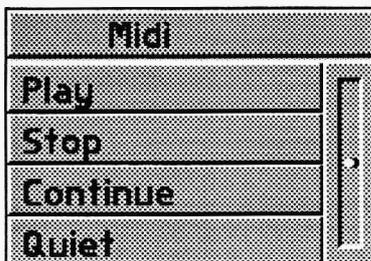
### ***Play***

On selecting this option, the MID files in the Audio directory are shown to you. Select a midi file to play. Once you do this, you will automatically advance to the next menu. Please note you have to make your midi files with a system other than TEMPRA Media Author before you try to play them with TEMPRA Media Author.

## Method

---

The Method menu offers you these selections.



### **Play**

This method allows you to play a midi file during a sequence.

### **Stop**

This method allows you to stop the midi file from playing back during a sequence. If you have stopped the midi playback,

you can restart it with a separate Continue line.

### **Continue**

This method allows you to restart the playback of a midi file that has been stopped with the Stop method.

### **Quiet**

This method allows you to quiet the playback of a midi file. You cannot restart the playback if you have used Quiet.

## Other Fields

---

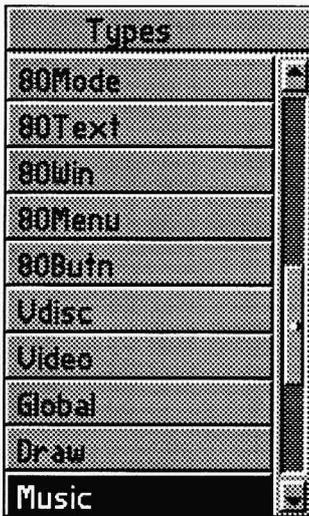
The fields that do not apply to this line are Sp/Dp and Parameters. Label, Next, and Wait are fields common to most lines. Apply them as necessary.

## Chapter Example

---

This example plays a file called text.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	MIDI	text	Play				



## Chapter 24

# ***A Music Event***

Music events control sound from the speaker found in every PC. No special hardware is needed to create Music events; you can create music using the keys on the keyboard as piano keys. This event can also play music while other events occur and create sound effects. The music files are ASCII text files with a suffix .MUS. The default directory for Music files is \TEMPRA\AUD. Be sure to place your music files in the default directory or change the directory using the menu under the Path button on the Menu Bar.

There is a chapter example sequence in the \TEMPRA\SEQ directory named MUSIC.SEQ. At the end of this chapter, there is a copy of the Workform and a brief description of each step. The rendition of "Flight of the Bumble Bee" is ours; you will see that we made each line a MB (Music Background) line and placed them to run consecutively. We then timed Buzz's rotations to go with the music background, using a 33 MHz computer. Depending on the speed of your computer, you may need to change the number of revolutions Buzz makes while each line of music plays. To do this, select the Parameters field on the lines showing AaPlay as the Method. The position box will appear, press Select to leave it where it is. The Number of Repeats? box appears. Type in a lower or higher number and press Select.

## **Type**

---

When the field cursor is in the event Type field and the menu pops up, highlight “Music” and press [Enter]. Your choice appears in the Type field.

## **File**

---

If the Music method that you specified requires a filename, you see a list of available Music files. Highlight a file and press [Enter]. If the Music file you want is not in the default directory, you can move the file to the default directory or change the Path, using the Path button on the Menu Bar.

## **Method**

---

This field is used to determine what kind of music you play.



### **Play**

Play musical notes using the musical information in the Parameters field.

### **Playfile**

Play musical notes using the information contained in a text file. When you select Playfile from the Music menu, the cursor goes to the File field and a list of files pops up. Music files are ASCII text files with a .MUS suffix. The default directory for Music files is \TEMPRAAUD.

Be sure to place your music files in the default directory or change the directory using the menu under the Path button on the Menu Bar. The Playfile method uses the text file lines specified in the Parameters field. Music can be played as a line event or in the background using the MB command in your text file. It is recommended that the MB command be used to play one line of text only. Approximately 215 characters make up one line.

### ***Kill***

Kill terminates any musical notes being played in the background, that is, music being played with the MB command, which allows the music to play while you show other events.

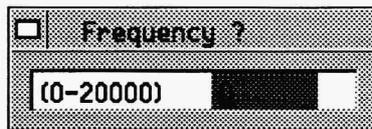
### ***Tone***

Play a tone with a frequency for the duration given in the Parameters field.

### ***Tonefile***

Play musical tones using the information contained in a text file. When you select Tonefile from the Music menu, the cursor goes to the File field and the list of files pops up. Music files are ASCII text files with a .MUS suffix. The default directory for Music files is \TEMPRA\AUD. Be sure to place your music files in the default directory or change the directory using the menu under the Path button on the Menu Bar. Tone text files are created using only the frequency (from 0-20000) and the duration (from 0-1000) numbers. An example would be 300 15 (300 frequency, 15 duration). This method uses the text file lines specified in the Parameters field.

### ***Freq***



Sets the speaker frequency and duration to the frequency and duration given in the Parameters field. With Freq, you must turn the speaker on and off using the Speaker On/Off menu. When you select

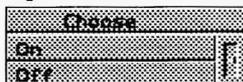
Freq from the Music menu, the cursor goes to the Parameters field.

### ***Freqfile***

When you select Freqfile from the Music menu, the cursor goes to the File field and a list of files pops up. Select the file you want to use. The default directory for Music files is \TEMPRA\AUD. Be sure to place your music files in the default directory or change the directory using the menu under the Path button on the Menu Bar. This method uses the text file lines specified in the Parameters field. With Freqfile you must turn the speaker on and off using the Speaker On/Off menu. Frequency text files are created using only the

frequency (from 0-20000) and the duration (from 0-1000) numbers. An example would be 300 15 (300 frequency, 15 duration). This method uses the text file lines specified in the Parameters field.

### **Speaker**



Turn the speaker on or off as specified in the Parameters column. The speaker must be turned on and off, using this choice, when Freq or Freqfile is used.

### **Loop**

Set the number of times for the next background music line to repeat. A background music line is a single Play Parameters field or a single PlayFile line that contains the “MB” command.

## **Parameters**

---

Parameters is the field where you type in your music notes or select the line of music in the text file you want to play.

### **Play**

This is the field where you type in the musical notes you want to play. Appendix D, Creating Music, contains information to help you create the notes you want to play.

### **Playfile**

The Parameters field specifies the starting and ending lines of the musical information to be played from the text file. When you select the text file, you are taken to the Parameters field to select the lines of music in the file that you want to play. First highlight the text line you want to start playing, then press [Enter]. Then highlight the text line you want to end playing and press [Enter].

### **Kill**

The Parameters field does not apply to the Music event with a Kill method.

## ***Tone***

When you select Tone from the Music menu, the cursor goes to the Parameters field and the Frequency box pops up. Type in the frequency you want. The Duration box pops up. Type in the duration you want. Tone turns the speaker on and off automatically.

## ***Tonefile***

The Parameters field specifies the starting and ending lines of the information in the text file to be played as tones. After you have selected your text file, the cursor moves to the Parameters field and the selected Tonefile pops up. First highlight the text line you want to start playing as tones, press [Enter]. Then highlight the text line you want to end playing, press [Enter].

## ***Freq***

The Frequency box pops up. Type in the frequency you want to use. The duration box pops up. Type in the duration. The duration figure tells the system how long to wait before bringing control back to TEMPRA Media Author. You must turn the speaker on using the Speaker selection on the Music menu.

## ***Freqfile***

After you select the file you want to use, the cursor goes to the Parameters field. The Parameters field specifies the start and ending lines of the information in the text file to be played as frequencies. First highlight the text line you want to start playing as tones, press [Enter]. Then highlight the text line you want to end playing, press [Enter]. The text file sets the speaker frequency to various frequencies, each for a specified duration, using the information contained in the text file. The duration pertains to the length of time from the start of the frequency to the time control returns to TEMPRA Media Author.

## ***Speaker***

When using Freq or Freqfile, you must turn the speaker on using this choice.

## ***On***

When you turn the speaker on, you hear the frequency that was last played.

Turn the speaker off to stop the frequencies.

### ***Loop***

Set the number of times for the next background music line to repeat. A background music line is a Play parameters field or a PlayFile line that contains the “MB” command.

### ***Other Fields***

---

The Label, Next, SP/DP, and Wait fields common to most lines. Apply them as necessary. For more information on these fields, please refer to the chapter on Common Fields.

### ***Chapter Example***

---

The following Workform shows an example of a Music event created using the computer keyboard.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Global		Colors		0 7B 1 0T 15 4H		
	Global	COUR25	Font				
	Video	MUSIC	Scrollup	4 E	152x 413y 234c 19w 10h nm	4 0/18	
	Music	MBBEE32	PlayFile		1 1		
	Video	TOUCAN	AaPlay	99 T	319bx 479by 6R		
	Music	MBBEE32	PlayFile		2 2		
	Video	TOUCAN2	AaPlay	99 T	0bx 479by 7R		
	Music	MBBEE32	PlayFile		3 3		
	Video	TOUCAN2	AaPlay	99 T	319bx 200by 7R		
	Music	MBBEE32	PlayFile		4 4		
	Video	TOUCAN2	AaPlay	99 P	3bx 200by 6R		
	Music	MBBEE32	PlayFile		5 5		
	BColor		Quick	99	Solid 0c		
	Video	TOUCAN2	AaPlay	99 P	155bx 362by 5R		
	Title		COUR25	P	The End. @249x 150y 95c	3 0/18	

The first three lines of this example set the Video mode, Global font colors, and the Global font choice.

The next line, a Video event, brings an ASCII text file to the screen. The file is named SCROLL.TXT and was created in our text editor. The Method used is ScrollUP. You will notice that the speed in Sp/Dp is set at 4. ScrollUP

requires a relatively slow speed to scroll smoothly and be readable by the viewer.

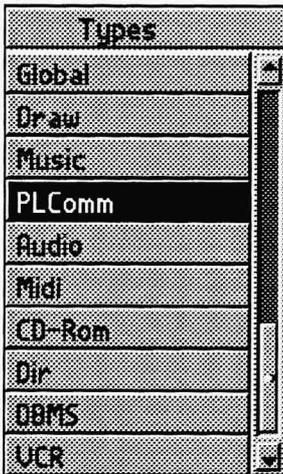
The music begins to play in a Music event using the Playfile Method. The file "MBBEE32.MUS was created in a text editor and saved as ASCII text. The Creating Music Appendix in this manual will help you create your own music files.

Now the Video event takes over with an AaPlay Method which brings animation to the screen using an animation file called TOUCAN2.FLI. The first two numbers in the Parameters field are the screen position numbers. The third number holds the Number of Repeats? choice.

We have created the music file in five MB (music background) lines and timed Buzz's revolutions to match the music. Depending on the speed of your computer, you may need to adjust the number of revolutions to get them to match the music.

The BColor event, Quick method is used to clear the screen for Buzz's last performance.

The last line is a Title event that brings "The End" to the screen. The Method field holds the font choice. You will notice the title text in the Parameters field along with the two numbers indicating the screen position and the color choice.



## *Chapter 25*

# ***A PLComm Event***

Use the PLComm event for multimedia as a smart manual in a factory. The PLComm event is specifically made for use with the Giddings & Lewis PiCLink system protocol. PiC stands for "Programmable industrial Computer." PiCLink systems are Programmable Logic Controllers used in industrial manufacturing environments. TEMPRA Media Author is used to monitor the status of the PLC and react according to the results of that monitoring.

### ***Type***

---

Select PLComm from the Type menu. We suggest that you be familiar with the Giddings & Lewis PiCLink system before you use this event. If your PiC has more than one memory bank, you need to know the bank number. You need to know the PiC internal variables, PiC memory locations and what they correspond to, and the relay numbers. You also need to know what to do with the information you get from this event.

## ***Method***

---

Following are the method choices from the PLComm menu.

### ***Config***

First you have to configure the communications between the PiC and the computer. This is not in the CONFIGMA, so it is done here. First, you set the serial port. You use either RS-232 port COM1 or COM2. Next, you set the baud rate to match the rate the PiC is using. Last, you set the memory bank number, if you have a PiCLink system with more than one bank. If your PiCLink system has only one bank, use one.

**Port: COM1 or COM2**

**Baud rate: 9600 or 1200**

**Bank: 1 or 2**

### ***ReadVar***

This method is used to assign PiC system variables to numeric variables for TEMPRA Media Author to use. These variables can come from the PiC, in which case they are preceded with "&". When you are reading a PiC variable, you have to tell the system what the variable number is according to the internal PiC absolute variable scheme.

You can also use a TEMPRA Media Author numeric variable to specify which PiC variable is to be read.

From the Parameters field, you are prompted to input a numeric var name for the variable that will hold the new information. You then choose whether to "Input PiC From?" "PiC Absolute" or "From Numeric Var." Choose "PiC Absolute" and enter the variable number, or choose "From Numeric Var" to get the value from an existing numeric variable.

### ***ReadMem***

This method is used to read PiC system memory to numeric variables for TEMPRA Media Author to use. The contents of the memory locations can come from the PiC, in which case they are preceded with "&". When you

are reading a PiC absolute memory location, you have to tell the system which memory location you are reading by using the internal PiC absolute memory addressing scheme.

You can use the TEMPRA Media Author variable to hold the address to be read.

From the Parameters field, you are prompted to input a numeric var name. You then choose whether to “Input PiC From?” “PiC Absolute” or “From Numeric Var.” You are then prompted for the PiC memory value or the name of the var.

### ***ReadCR***

This method is used to read PiC control relays to numeric variables for TEMPRA Media Author to use. These values for the variables can come from the PiC control relays, in which case they are preceded with “&”. When you are reading a PiC control relay you have to tell the system which control relay you are reading by using the internal PiC control relay numbering scheme.

You can also use a TEMPRA Media Author variable to hold the relay number. The values coming in from the control relays are 0 for off and 128 for on.

From the parameters field, you are first prompted to input a numeric var name. You then choose whether to “Input PiC From?” “PiC Absolute” or “From Numeric Var.” If you choose “PiC Absolute,” enter the control relay number which you want read. If you choose “From Numeric Var,” enter the variable name that holds the value you want.

### ***Version***

This method sets the version number for the COM409 into a TEMPRA Media Author numeric variable. You are prompted for the numeric var name.

## **Trap**

Trap methods poll the serial port at least once per line, usually at the end of the line. Trap methods work only one at a time, so you reset any active trap method with a subsequent trap method. The Trap methods use the following “Test” and branch to a subroutine or jump to a label if the test is true.

==	Equals
!=	Does not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

## **TrapSUBV**

The TrapSUBV method traps a PiC system var and compares it to either a literal value or a numeric var. If the test is “True,” it calls the subroutine designated in the Next field.

From the Parameters field:

- Input** the PiC system var name.
- Choose** the kind of test.
- Compare** to a literal value or a numeric var.

## **TrapSUBM**

The TrapSUBM method traps a PiC system memory location value and compares it to either a literal value or a numeric var. If the test is “True,” it calls the subroutine designated in the Next field.

From the Parameters field:

- Input** the PiC system memory location.
- Choose** the kind of test.
- Compare** to a literal value or a numeric var.

**TrapSUBR**

The TrapSUBR method traps a PiC system relay value and compares it to either a literal value or a numeric var. If the test is “True,” it calls the subroutine designated in the Next field.

From the Parameters field:

**Input** the PiC system relay number.

**Choose** the kind of test.

**Compare** to a literal value or a numeric var.

**TrapJMPV**

The TrapJMPV method traps a PiC system var and compares it to either a literal value or a numeric var. If the test is “True,” it branches to the Label entered in the Next field.

From the Parameters field:

**Input** the PiC system var name.

**Choose** the kind of test.

**Compare** to a literal value or a numeric var.

**TrapJMPM**

The TrapJMPM method traps a PiC system memory location value and compares it to either a literal value or a numeric var. If the test is “True,” it branches to the Label entered in the Next field.

From the Parameters field:

**Input** the PiC system memory location.

**Choose** the kind of test.

**Compare** to a literal value or a numeric var.

### TrapJMPR

The TrapJMPR method traps a PiC system relay value and compares it to either a literal value or a numeric var. If the test is “True,” it branches to the Label entered in the Next field.

From the Parameters field:

**Input** the PiC system relay number.

**Choose** the kind of test.

**Compare** to a literal value or a numeric var.

### TrapOFF

This turns the PiC input trapping off.

## Next

---

In the Next field, enter the name of the subroutine or the Label to branch to. It is important to note that if using the TrapSUB methods for a PLComm event, when the subroutine is executed, control is returned to the line in the sequence following the event that was taking place at the time the subroutine was called, not necessarily the line following the PLComm event. Since the Trap methods start polling the serial port, and then execution of the sequence continues with subsequent events, it could be several events later in the sequence before the Trap test becomes “True.”

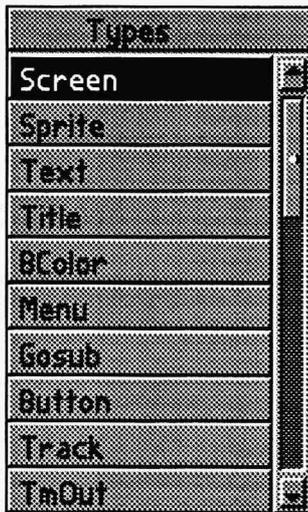
## Chapter Example

---

This example traps PiC variable 88 and tests it against the value in #temp. When the test is true, control goes to the subroutine labeled Err2.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	PLComm		TrapSUBV		488  - #temp		Err2

The test occurs during the sequence as other events are executing by running the test after each event. When the test is true, the subroutine occurs no matter where you are. After the subroutine finishes, control returns to the line after the line that was running when the test was true.



## Chapter 26

# A Screen Event

Bring PCX and TGA images onto your screen. Scale, position, show your pictures with a wide variety of wipes.

### **Type**

---

To show a full frame or partial screen graphic or converted digitized video image on the screen during your sequence, select Screen from the Type menu. A Screen event remains on your video monitor until another Screen or BColor event. You can place a scaled image over a full screen image, just be sure the palettes are the same.

### **File**

---

A list of available images appears. TEMPRA Media Author automatically shows the files that are consistent with your video mode; i.e., PCX in 256-color mode, TGA in high-color mode. See the chapter on Video (Set Mode) to reset the resolution. See the Global event chapter in this manual for more information regarding switching between PCX and TGA graphics file formats and how to use them. Select the image file you want, your choice appears in the File field, the highlight goes to the Method field, and the Wipe menu appears.

## Method

---

The menu that appears lists ways to bring your screen event into the presentation. These “wipes” are similar in function and description to wipes used in the broadcasting industry. Select one of these wipes and TEMPRA Media Author will move on to the next menu.

## Parameters

---

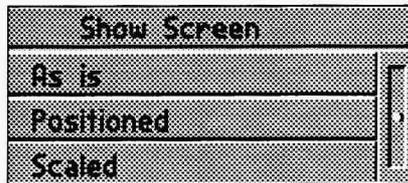
The Parameters field is used when bringing in a Screen that is to be scaled or positioned.

### As Is

This feature brings the image onto the video screen without modification. No entry appears in this column because this is the default. Once the image is loaded to the screen, the entire visible screen may be seen by selecting the Remove Menu button on the Menu Bar. To return to the menu screen, press Select.



**Sharp Boy says:**  
You don't have to bring in a complete screen. Sometimes you may want just a small part of an image and the Screen event lets you do that. This can save a lot of disk space and make your demos easier to distribute.



### Positioned

This selection allows you to pick a location for the image you are bringing onto the screen. You choose a location for the placement of the upper left corner of the image by moving the screen cursor and pressing [Enter]. This selection is useful when you have a small graphic you want to position on the screen.

### Scaled

This selection allows you to bring images onto the visible screen and scale them to the size you want. Prompts appear in the upper left corner telling you to position the lower left corner using the cursor keys. You can also use the mouse by moving the cursor to the lower left position, pressing [Enter], then expanding the box to the desired size and pressing [Enter] again. During the editing stages, you can be more precise by moving the cursor over the parameters field and pressing the [F3] key. You can then type the exact position you want into the field. You can also use the arrow keys on the keyboard for precision placement.

## Other Fields

Label, Sp/DP, Wait For, and Next are events common to most lines and are discussed in the Common Fields chapter. Apply these events as necessary.

## Chapter Tutorial

This chapter tutorial is named SCREEN.SEQ in the \TEMPRA\SEQ directory. Following is a copy of the Workform and the steps necessary to recreate the sequence.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Screen	TUTOR1	Dissolve	99		3 0/18	
	BColor		Tp_Bot	99	Solid 62c		
	Screen	TUTOR2	Bot_Tp	99	260bx 318by	3 0/18	
	BColor		Diag_UL	99	Solid 133c		
	Screen	SLIDE20	Rt_Lft	99	163bx 125by 476ex 377ey	3 0/18	

*Line 1 sets the video mode so you can use the PCX images in this tutorial.*

### Recreating Line 1

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

*Line 2 contains a screen size PCX graphics image that dissolves onto the screen "as is."*

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK** to accept.
5. **Parameters** pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The image appears on the screen; the Line Counter advances to Line 3.

*Line 3 contains a background color that wipes onto the screen from top to bottom.*

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select  **Tp\_Bot**.
3. The **SD/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. From the menu, select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen color changes and the Line Counter advances to Line 4.

*Line 4 contains a full screen PCX image that is "positioned" on the screen.*

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR2.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Bot\_Tp**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **Positioned**. A box will appear in the upper left corner and a cross will

appear on the screen. Use the cursor keys or mouse button to position the upper left corner where you want it. Press [Enter]. We are positioning a full-screen graphic to show you how Positioned works. Positioned allows you place a small graphics image anywhere on the screen.

6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The image appears on the screen; the Line Counter advances to Line 5.

*Line 5 contains a background color that wipes onto the screen diagonally.*

### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Diag\_UL**.
3. The **SD/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **Yes**. A cross-shaped cursor appears on the screen. Move the cursor around on the screen. Notice how the box changes color to reflect the current color the cross is positioned over. Pick a color from the image and press **Select**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen color changes and the Line Counter advances to Line 6.

*Line 6 contains a full screen PCX file we "scaled" into the box we set up in this event.*

### **Recreating Line 6**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **SLIDE20.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Rt\_Lft**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **Scaled**. A box will pop up in the upper left corner and a cross will appear on the screen. Position the lower left corner where you want it and press [Enter]. Now size the box by moving the mouse. When the size is correct, press [Enter].

6. The **Wait** menu pops up. Select **Any Key**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The graphic appears on the screen; the **Line Counter** advances to **Line 7**.
9. Rewind to **Line 1** and press **F6** to see the sequence play.



## Chapter 27

# A Sprite Event

A sprite is a picture clip that moves across the screen. This is a simple animation event that adds motion to presentations. Move sprites around on your screen. Make them appear and disappear. Scale them; reuse them.

### **Type**

---

Select Sprite from the Type menu to bring a movable (or moving) graphic or digitized picture clip into your sequence. Sprites are generally small relative to the screen size and carry a PCC extension in the 8-bit mode or a WIN extension in the 16-bit and 24-bit modes.

### **File**

---

Highlight a file on the pop-up file list and press [Enter]. Your choice appears in the File field. TEMPR Media Author lists only the image files that can be used as sprites. Any image that is small enough and saved as an 8-bit PCC file can be used as a sprite. In 16- and 24-bit true-color modes, the file extension becomes WIN. TEMPR Media Author, by default, looks for a PCC sprite file. To change the default, use a Global event and select SPR\_WIN.

### ***Method***

---

The method determines how your sprite appears on the screen before it moves. There are three ways to bring a sprite onto the screen.

#### ***Simple***

To get the sprite to just appear without motion, select Simple.

#### ***Wipe In***

When you want to wipe the sprite onto the screen, select Wipe In. The Wipe menu pops up. Select a wipe from the menu.

#### ***Scale***

Use this to scale the sprite before you show it on the screen. This method allows you to make a sprite any size your Sprite buffer can hold. The size of the Sprite buffer varies according to system resources, if you have not set it with the `Sprite_Buf` environmental variable explained in the Technical Information appendix of this manual. After you choose the Scale method, the sprite and a screen instruction appear. If you try to scale the sprite to a size that the system cannot accommodate, a message appears telling you the sprite is too large. When you press [Enter], the percentage of the scaling is entered into the Parameters field. The scaling is always proportional.

### ***Parameters***

---

Once you select a sprite, it appears in the lower left corner of your screen. A prompt asks you to “Use your cursor keys to move the sprite to its beginning position, then Press [Enter].” Use the mouse to position the sprite at its beginning position and press Select. This is the position where you want the sprite to first appear. If you have set an active Viewport, and do not see the sprite, it may be out of the Viewport area. Press the space bar to see the active Viewport as a square and to reveal the sprite. For information on the Viewport method, see the Video chapter in this manual.

A second prompt now asks you to “Use your cursor keys to move the sprite to its ending position. Then Press [Enter].” With the mouse, move the sprite

to the ending position. The sprite remains on your video monitor. Pressing [Ctrl X] shows the x,y coordinates in the bottom left of your screen. Your cursor moves to the Wait field and the Wait Menu pops up.

## Other Fields

Label, Next, Sp/Dp, and Wait are fields common to most lines. Apply them as necessary. For more information on these fields, see the chapter on Common Fields.

## Special Features of Sprites

Sprites can add a special flair to presentations that might otherwise be static. Even if you don't have any animation files to use (with .FLC and .FLI extensions), you can put life into presentations very simply.

### Creating Sprites

Be sure to make your Sprites with the same number of available colors as the images they will move over. Sprites to be shown with 256-color images must be made in the 256-color mode, 16-color images require 16-color sprites, and so on. If palettes are different, the colors on the sprite will change to reflect the colors in the image. Working with the same palette on 8-bit images and 8-bit sprites provides a consistent palette for more pleasant-looking presentations.



Sharp Boy says:  
Avoid leaving trails of sprite pieces like those shown at the right. A sprite with a 10 pixel border will leave trails if they are moved at speeds over 10.

The border around your Sprite must be transparent black. So if you want to move a sprite at speed 12 you should have a border of at least 12 pixels on that sprite.

### Creating a Sprite From Scratch

Use TEMPRA PRO, TEMPRA GIF, or your drawing program to create a rectangle that has only the transparent color in it (color value of 0). This color is black in the default palette. Draw your sprite inside this rectangle. Leave "black space" around the sprite, and save it to the disk.



## **Creating a Sprite From an Existing Image**

Use TEMPRA PRO, TEMPRA GIF, or your drawing program to load an image. Put a 0 color (black) rectangle around the part of the image that you want to be a sprite. Erase everything in the rectangle that is not part of the sprite you want to see. Erasing really means turning all the pixels to color 0 (black). Save the portion of the screen that is within the black rectangle.

## **Chapter Tutorial**

---

SPRITE.SEQ in the \TEMPRA\SEQ directory shows the three ways of including a Sprite event in your sequence. Following is a copy of the Workform and the steps necessary to recreate the sequence. To just create the sprite, do only Line 4 in this tutorial.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Global		SPR_PCC				
	Video		SetMode		SGA_2E 640x480 256		
	Bcolor		Tp_Bot	99	Solid 75c		
	Sprite	HAND		5 E	-20bx 236by 261ex 222ey		
	Sprite	HAND	Hrz_In	5 E	190bx 419by 238ex 82ey		
	Sprite	HAND	138*	5 E	226bx -12by 366ex 366ey		

*Lines 1 and 2 are necessary for the sprite we have used in the tutorial to show correctly on your screen. If your default matches these two lines, you can skip them.*

*Line 1 contains a Global event to select PCC sprite files.*

### **Recreating Line 1**

1. Select the box under **Type**. From the pop-up menu, select **Global**.
2. From the **Type** menu, select **SPR\_PCC**.
3. Select **Don't Wait** from the **Wait** menu.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

*Line 2 contains a Video event, Set Mode method.*

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the Set Mode menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 3.

*Line 3 creates a background color that wipes onto the screen from top to bottom.*

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Tp\_Bot**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen color changes; the Line Counter advances to Line 4.

*Line 4 brings the sprite onto the screen using the "simple" method.*

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **Sprite**.
2. The **File** menu pops up. Select **HAND.PCC**.
3. The **Method** menu pops up. Select **Simple**.
4. The **SP/DP** menu pops up. Set the speed at **5**. Select **Erase**. Select **OK**.
5. The **Parameters** menu pops up. Position the hand's starting point off-screen on the left side. Then position the ending point where you want it on the screen.
6. The **Wait** menu pops up. Select **Don't Wait**.

7. The **Next** menu pops up. Press **Select** to accept the default.
8. The hand moves to the ending point; the Line Counter advances to Line 5.

*Line 5 brings the sprite onto the screen using the “wipe in” method.*

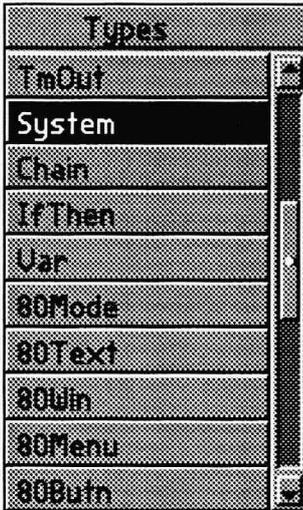
### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **Sprite**.
2. The **File** menu pops up. Select **HAND.PCC**.
3. The **Method** menu pops up. Select **Wipe In**. The **Wipe** menu pops up. Select **Hrz\_In**.
4. The **SP/DP** menu pops up. Set the speed at **5**. Select **Erase**. Select **OK**.
5. The **Parameters** menu pops up. Position the hand’s starting point at the top of the screen. Position the ending point where you want it on the screen.
6. The **Wait** menu pops up. Select **Don’t Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The hand moves to the ending point; the Line Counter advances to Line 6.

*Line 6 brings the sprite onto the screen using the “scale” method.*

### **Recreating Line 6**

1. Select the box under **Type**. From the pop-up menu, select **Sprite**.
2. The **File** menu pops up. Select **HAND.PCC**.
3. The **Method** menu pops up. Select **Scale**. The hand appears on the screen in a box. Use the mouse to enlarge the hand. Press **[Enter]**.
4. The **SP/DP** menu pops up. Set the speed at **5**. Select **Erase**. Select **OK**.
5. The **Parameters** menu pops up. Position the hand’s starting point at the bottom of the screen. Then position the ending point where you want it on the screen.
6. The **Wait** menu pops up. Select **Don’t Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The hand moves to the ending point; the Line Counter advances to Line 7.
9. Rewind to Line 1 and press **F6** to play the sequence.



## Chapter 28

# A System Event

Use TEMPRA Media Author as a program manager, to start your own programs, or secretly run batch files and utilities as part of your presentation. Select this event to execute programs stored on your hard drive or to call DOS functions, batch files, and commands mid-sequence. Typically, TEMPRA Media Author keeps only 15k for itself and the rest of your memory is available for your programs. Before you can use a System event, you must enter the following line in your CONFIG.SYS file and reboot your computer after this line is added:

```
SHELL=C:\COMMAND.COM C:\ /P /E:1024
```

### **Type**

---

Select the Type field, and the event menu pops up. Select System and press [Enter] or the left mouse button.

### **Method**

---

Choose the amount of memory you need in the shell: ALL, 100K, 200K, 300K, or 400K.

## ***Parameters***

---

After you have chosen the System type, a prompt asks you to “Input DOS Parameters.” The entry you make is executed by the computer as if you had entered it from the system prompt. After the program you named in the Parameters field runs, control passes to the sequence once again.

## ***Other Fields***

---

Label, File, SP/DP, and Next are Fields common to most lines. Apply them as necessary. For more information on these fields, refer to the chapter on Common Fields.

## ***Chapter Example***

---

This example tells TEMPRA Media Author to go to the system and get a directory.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
sysye	System			ALL	Dir		



## Chapter 29

# A Text Event

Put text on the screen in numerous fonts, colors, and methods. Save on disk space and increase flexibility by using ASCII files for the text.

### **Type**

Select the Text event from the Type menu. Text events show text from any word processing program saved as an ASCII file. You can display text in TEMPR Media Author in a variety of ways: in any graphic mode; font or color; in a frame; or directly over an image.

### **File**

A list of Text files with the .TXT extension appears. Highlight the file you want and press Select. Text files can be written in the TEXTEDIT utility program supplied with TEMPR Media Author, or with any word processing program and saved as an ASCII file with a .TXT extension. Text is selected one line at a time, so your text files can be as short as one line or as many lines as you want. Text files must be located in the default text directory, which is \TEMPRA\TXT, unless you change the path for your text files using the Path button on the Menu Bar.

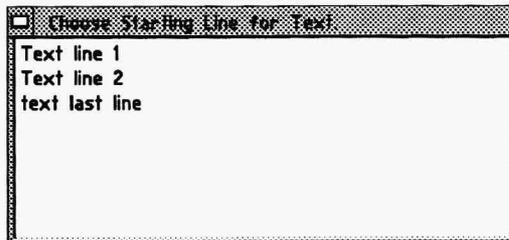
## Method

---

A font choice menu is the method of this event. Choose from the default, VV fonts or KST fonts. When you choose VV or KST fonts, you see a list of fonts. Choose one and proceed to the Display menu. Choosing the default font allows you to use the font set as the default in the Global event. In other words, the default font can be any font you set as the default using a Global event, selecting Font from the Global menu and then selecting the default font.

## Sp/Dp

---



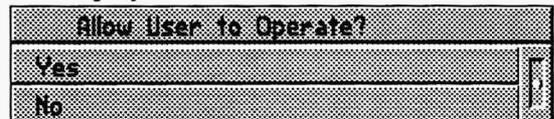
The first decision of the display menu is Permanent, Temporary, or Erase display types. Choose one based on how long you want the text to display. Review the section on Sp/Dp (Speed/Display) in the Common Fields chapter of this manual for further information.

If your text file contains more than one line, choose the line in the file to start your text display. TEMPRA Media Author will display the line you select and everything from that line on. If you have a long text file, use the cursor keys to scroll through the file and find the line you want.

## Parameters

---

The first menu to appear is the "Allow User to Operate?" menu. Selecting Yes at this menu sets the text to display in a framed box with scroll bars so the viewer can see the text file within the presentation at their leisure. Selecting No at this point presents your text framed or unframed. The next pop-up menu asks if you want to draw a frame. Answering Yes will place your text in a framed box. A No answer will place the text directly over an image or background color. The next menu to appear asks you to draw a frame around the section of text you have selected. Then you position the text on the screen.



The Parameters field is where you determine the color of the text and whether you want a frame around it. You set the size and location of the window. When you are creating the Text event for the first time, you are given all the set up choices in order. When you are editing, you are presented with a menu asking which of the options you want to modify.

### ***All***

This choice takes you through the steps as if you were writing the line for the first time.

### ***Text Color***

You are asked if you want to select the color from the graphics screen. If you answer Yes, a cursor appears that has a square border. The cursor's border color is the selected color. Move the cursor over the color you want to select and press [Enter]. If you want a color from the color palette, select No, and a color menu appears. Choose the color you want from the small squares and select OK.

### ***Operate/Frame***

The "Allow User to Operate?" menu pops up. Selecting Yes allows the user to operate the text window by scrolling through it with the cursor keys. It also allows the user to pan right and left to see long lines of text that do not fit into the text window. Selecting No allows you to choose framed or unframed text from the "Draw Frame?" menu.

### ***Not Framed***

If you want the text to display over an image without a frame, then select No from the "Draw Frame?" menu.

### ***Framed***

If you want the text in a framed window answer Yes to the "Framed?" question. A framed window has the color attributes set with the Global event. The Global values for the windows at start up are the same as the Workform. To change the border, background, and highlight colors of a framed window, select Global from the Type menu; then select Colors from

the Global menu. A box appears in the upper left corner of the screen. You can make your choices from a graphic displayed on the screen or from the color palette.

### ***Width/Height***

Once the operation question is answered for framed text windows, you must set the size of the window. The text from the file is shown on the screen in the font you chose. Use the cursor keys or mouse to stretch the frame around your text. You can adjust the placement of text within the frame by placing hard returns before and/or after the text you want to display, using your text editor.

### ***Justify***

You have the ability to see the text with left or right justification, or centered. Remember that TEMPRA Media Author thinks spaces are characters, so if you have spaces at the end of your text lines, the centering will not look correct. If this is the case, go back into the text file from your text editor and delete the spaces at the end of the lines.

### ***Position***

Now that the text display options have been set, it is time to place the text on the screen. The Text is seen only when it is within the active Viewport. If you have not set an active Viewport, the entire screen is the active Viewport. See the Video event chapter, Viewport method, for more information on Viewport. If you cannot see the Text to position it, press the space bar to reveal the Viewport as a square, and to show your text. Your computer screen prompts you to “Use your cursor keys to position your Text. Then press [Enter].” To see the x,y coordinates as you move the text, press [Ctrl] X. Use your cursor keys or mouse to position your Text and press [Enter]. The Text coordinates appear in the Parameters field. Your highlight cursor moves to the Wait field and the Wait menu pops up.

### ***Other Fields***

---

Label, Next, and Wait are fields common to most lines. For more information on these fields, see the Common Fields chapter in this manual.

## Chapter Tutorial

This chapter tutorial is named TEXT.SEQ in the \TEMPRASEQ directory. Following is a copy of the Workform and the steps necessary to recreate the sequence. This sequence is designed to show you most of the Text event features. We have provided brief explanations so you can use the instructions to create your own sequences.



Sharp Boy says:

We've given you lots of examples of Text events. Play the sequence first to see how the Text event can help get your message across. The italicized sentences tell you how to create the lines; so pick and choose what you want to try.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Video	TUTOR1	LoadPal				
	Global		Colors		6 102B 40 58T 49 27H		
	BColor		Box_in	99	Solid 197c		
	Text	TEXT	HELV20	P 001	56x 388y 10c 16w 3h nnm	2 0/18	
	Text	TEXT	HELV15	P 003	184x 309y 179c 30w 3h nnm	3 0/18	
	Text	TEXT	BLOCK30	P 005	49x 248y 47c 6w 3h nnm	1 0/18	
	Text	TEXT	ASC10	P 007	267x 207y 0c 4w 3h nnm	1 0/18	
	Text	TEXT	COUR25	P 009	370x 144y 55c 13w 3h nnm	2 0/18	
	BColor		Hz_Out	50	Solid 216c		
	Text	TEXT	COUR20	P 011	36x 405y 85c 47w 3h nnm	2 0/18	
	Text	TEXT	COUR20	P 013	102x 274y 102c 5w 3h nnm	1 0/18	
	Text	TEXT	COUR20	P 015	185x 206y 41c 5w 3h nnm	1 0/18	
	Text	TEXT	FUTURA20	P 017	279x 297y 11c 5w 3h nnm	1 0/18	
	Text	TEXT	HELV30	P 019	374x 191y 191c 7w 3h nnm	2 0/18	
	Screen	TUTOR1	Dissolve	50			
	Text	TEXT	ASC20B	P 021	110x 370y 127c 24w 3h nnm	2 0/18	
	BColor		Tp_Bot	50	Solid 254c		
	Text	TEXT	TIMES25	P 023	138x 327y 12c 14w 3h nfm	2 0/18	
	BColor		Hz_in	50	Solid 161c		
	Text	TEXT	FUTURA20	P 025	83x 319y 61c 22w 7h ofl		
	BColor		Box_In	50	Solid 133c		
	Text	TEXT	ASC15B	P 030	86x 385y 61c 28w 5h nfm	2 0/18	

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Text	TEXT	HELV15	P 032	162x 305y 82c 23w 5h nfm	1 0/18	
	Text	TEXT	BLOCK20	P 034	122x 227y 0c 23w 5h nfm	4 0/18	
	BColor		Box_Out	50	Solid 247c		
	Text	TEXT	COUR20	P 036	43x 387y 61c 44w 5h nfm	2 0/18	
	Text	TEXT	COUR20	P 038	45x 260y 61c 7w 5h nfl	1 0/18	
	Text	TEXT	COUR20	P 040	473x 255y 61c 8w 5h nfr	1 0/18	
	Text	TEXT	COUR20	P 042	221x 212y 62c 13w 5h nfm	1 0/18	
	BColor		Lft_Rt	50	Solid 18c		
	Text	TEXT	HELV20	P 045	51x 397y 92c 26w 12h nml	1 0/18	
	Text	TEXT	BLOCK20	P 056	113x 185y 55c 24w 3h nmm		

*There are a number of lines that just set the stage for the text events. Since users with all kinds of hardware will be viewing the sequence, we have added lines 1, 2 and 3 so all can view and learn from watching. When you are creating on your own machine, these lines may not be necessary.*

### **Recreating Line 1**

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. The **Method** menu pops up. Select **LoadPal**.
3. The **Files** menu pops up. Select **TUTOR1.PCX** file. This will set the 256-color palette for the rest of the tutorial.
4. The **Wait** menu pops up. Select **Don't Wait**.

5. The **Next** menu pops up. Press **Select** to accept the default.
6. The Line Counter advances to Line 3.

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **Global**.
2. From the **Global** menu, select **Colors**. The **Colors** menu will pop up.
3. **Select** the colors you want for the framed windows from the palette. When you are finished, select **Done**.
4. The Line Counter advances to Line 4.

*Line 4 creates the background color for the first five Text events.*

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Vrt\_In**.
3. The **SP/DP** menu pops up. Press the **Up** arrow until the speed is **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 5.

*Lines 5 - 9 show you some of the fonts available in TEMPRA Media Author and are intended to familiarize you with selection of fonts.*

### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the first line in the text file, **Welcome to the Text Event**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select

- Middle.** For **Choose Color From Image?**, select **No**. Choose a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
  7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
  8. The **Next** menu pops up. Press **Select** to accept the default.
  9. Text appears on the screen; the Line Counter advances to Line 6.

### ***Recreating Line 6***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **You can choose....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Choose a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **3** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 7.

### ***Recreating Line 7***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.

3. The **Method** menu pops up. Select **KST Fonts**. Select a large typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Large**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Choose a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 8.

### ***Recreating Line 8***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a small typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Small**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Choose a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.

8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 9.

### ***Recreating Line 9***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **medium** typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Or in between**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Choose a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of **0** on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 10.

*Line 10 creates the background color for the next five Text events.*

### ***Recreating Line 10***

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Hrz\_Out**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 11.

---

*Lines 11 - 15 show you how the choice of color can brighten up your presentations and introduce you to the color palette.*

### **Recreating Line 11**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **You can choose any color....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 12.

### **Recreating Line 12**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Red**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a red square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen

to be positioned. Move the text to where you want it on the screen and press **Select**.

7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 13.

### ***Recreating Line 13***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Blue**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a **blue** square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 14.

### ***Recreating Line 14***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Green**. For **Allow User to Operate?**, select **No**. For

---

**Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a **green** square from the palette and select **OK**.

6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 15.

### ***Recreating Line 15***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, or **Yellow**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a **yellow** square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 16.

*Line 16 is a Screen event to bring in an image for Line 17.*

### **Recreating Line 16**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For **Show Screen**, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The image appears on the screen; the Line Counter advances to Line 17.

*Line 17 shows you how to place text over a graphic image.*

### **Recreating Line 17**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **You can write....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **Yes**. A cross-shaped cursor will appear on the screen. Move the cursor over the image until you find the color you want to use, press **Select**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 18.

*Lines 18, 20, 22, 26 and 31 are all BColor events that provide background color for the Text events that follow them.*

### **Recreating Line 18**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Tp\_Bot**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 19.

*Line 19 introduces framed text and how to place it on the screen.*

### **Recreating Line 19**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Or frame it....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 20.

### ***Recreating Line 20***

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Hrz\_In**.
3. The **SP/DP** menu pops up. Press the **Up arrow** until the speed is **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 21.

*Line 21 shows how you can use framed text and let the viewer/user participate by scrolling through the text and making a selection.*

### ***Recreating Line 21***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **You can let the viewer....** For **Allow User to Operate?**, select **Yes**. For **Draw Frame**, select **Yes**. For **Justify**, select **Left**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Don't Wait**. (The viewer must press [Enter] or **Select** to continue.)
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 22.

## **Recreating Line 22**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Box\_In**.
3. The **SP/DP** menu pops up. Press the **Up arrow** until the speed is **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 23.

*Lines 23 - 25 demonstrate how you can size and place your text.*

## **Recreating Line 23**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **You can adjust the width....** For **Allow User to Operate**, select **No**. For **Justify**, select **Center**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 24.

### ***Recreating Line 24***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **by adjusting the length....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Center**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 25.

### ***Recreating Line 25***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **large** typeface.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **and the font size....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Center**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.

7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 26.

### ***Recreating Line 26***

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Box\_Out**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 27.

*Lines 27 - 30 show you the text justification possibilities for your presentations.*

### ***Recreating Line 27***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **large** typeface.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **And, you can justify....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Center**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **2** and press **Select**. Press **Select** again for default of 0 on 18th seconds.

8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 28.

### ***Recreating Line 28***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **large** typeface.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Left**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Left**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default of 0 on 18th seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 29.

### ***Recreating Line 29***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **large** typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Right**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Right**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to

be positioned. Move the text to where you want it on the screen and press **Select**.

7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 30.

### ***Recreating Line 30***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a **large** typeface.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Middle**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text to where you want it and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 31.

### ***Recreating Line 31***

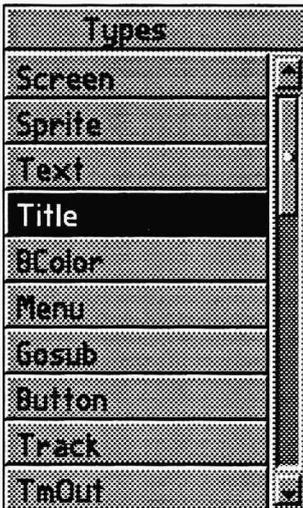
1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the **Wipe** menu, select **Lft\_Rt**.
3. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
5. The **Wait** menu pops up. Select **Don't Wait**.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen changes color; the Line Counter advances to Line 32.

### ***Recreating Line 32***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **As you can see....** For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Left**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait for Timeout**. Select the box under **Wait**, type **1** and press **Select**. Press **Select** again for default.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 33.

### ***Recreating Line 33***

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TEXT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
5. **Parameters** pops up. For **Choose Starting Line for Text**, select the line in the text file, **Thanks for watching. Enjoy**. For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. The color palette appears. Choose any color square from the palette and select **OK**.
6. Framed text appears in the top left corner. Move the frame around the text you want to display and press **Select**. The text appears on the screen to be positioned. Move the text where you want it on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Don't Wait**.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on the screen; the Line Counter advances to Line 34.
10. Rewind the sequence to Line 1, press **F6** to see it play.



## *Chapter 30*

# *A Title Event*

For small captions, titles and screen notes, use the Title event. This is similar to a Text event, but allows you to type right in the program and not with a text editor or word processor.

### ***Type***

---

Select this event to place a title with the font and color of your choice in your sequence. Titles are input while a sequence is being built. A title cannot exceed 9 characters in length. You can have longer titles by creating several Title events in a row and positioning them together. We did this in Lines 7, 8 and 9 in the chapter tutorial.

### ***Method***

---

The cursor moves to the Method field and the Title Font Menu appears. Choose the default font, a VV font or a KST font. If you choose VV font or KST font, a menu of those fonts pops up. Select a font. Your choice appears in the Method field.

### **Parameters**

---

The parameters field is where you input the title text, set the color and position on the screen.

#### **Create New Titles**

The cursor moves to the Parameters field and the prompt “Input Title” appears. Titles may be up to 9 alphanumeric characters in length. Type a title and press select or [Enter].



Sharp Boy says:  
If you have a few words to say, use the Title event to say them. You can string several Title events together and save yourself the time of making a text file.

You can create titles as long as you wish by creating subsequent Title events and positioning them accordingly. An example of this technique is seen in lines 7 through 9 in the chapter tutorial. If you are thinking of making a long title, you might want to consider using a Text event.

Now you are asked to choose the color from the screen or the palette. If you choose to pick a color from the screen, you see a small cursor with a colored box on top of your image. As you move the cursor, you see that the color in the box changes. It is showing you the color of the pixel on the screen under the cursor. When you press [Enter], the Title text will be the color under the cursor.

If you choose not to pick the color from the screen, a color palette appears. Move the cursor over the color you want and press Select and then OK.

TEMPRA Media Author now clears the screen. The title, with the options you've specified, appears in the lower left corner. The title only shows up in the active Viewport. If you cannot see the title, it may be because the title is outside the active Viewport. Press the space bar to reveal the active Viewport as a square, and to show your title. See the Video chapter, Viewport method, in this manual for more information on Viewport.

Your computer screen prompts: “Use your cursor keys to position your title. Then Press [Enter].” Titles only have one set of coordinates. Press [Ctrl] X to see the x,y coordinates on the graphic screen status line. The x,y coordinate message will change as your title moves.

Use your mouse or the cursor keys to position the title and press Select or [Enter]. The title coordinates and color number appear in the Parameters field.

### ***Editing Title Events***

When you position the field cursor over the Parameters field of a Title event and press Select or [Enter] you can see a menu giving you choices to edit a single attribute of the Title event or all of them.

#### ***All***

Selecting this title editing option recreates the process of making the title as if you were doing it for the first time.

#### ***Title***

Selecting this option allows you to edit only the title itself. The prompt “Input Title” appears. Titles may be up to 9 alphanumeric characters in length.

#### ***Position***

Your computer screen prompts: “Use your cursor keys to position your title. Then Press [Enter].” Reposition your title.

#### ***Color***

Selecting this option lets you change only the color of the title text.

### ***Other Fields***

---

Label, SP/DP, Wait, and Next are fields common to most lines. Apply them as necessary. For more information on these fields, refer to the Common Fields chapter in this manual.

## Chapter Tutorial

This Chapter Tutorial is name TITLE.SEQ in the \TEMPRA\SEQ directory. Following is a copy of the Workform and the steps necessary to recreate the sequence.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Video	TUTOR1	LoadPal				
	Screen	TUTOR1	Hzz_In	99			
	Screen	TUTOR2	Tp_Bot	99			
	Title		BLOCK20	P	Hello! @95x 410y 190c	1 0/18	
	Title		BLOCK20	P	I'm Buzz. @69x 375y 249c	2 0/18	
	Title		HELV20	P	Hope you @209x 70y 92c		
	Title		HELV20	P	enjoyed @312x 70y 163c		
	Title		HELV20	P	our show! @401x 70y 163c	3 0/18	

*Lines 1 and 2 set the stage for the Title events to follow. When you are creating on your own machine, these steps may not be necessary.*

### Recreating Line 1

1. Select the box under **Type**. From the pop-up menu, select **Video**.
2. From the **Video** menu, select **Set Mode**. From the **Set Mode** menu, select **SGA\_2E 640 x 480 256**.
3. The **Wait** menu pops up. Select **Don't Wait**.
4. The **Next** menu pops up. Press **Select** to accept the default.
5. The Line Counter advances to Line 2.

### Recreating Line 2

1. Select the box under **Type**. From the pop-up menu, scroll down and select **Video**.
2. The **Method** menu pops up. Select **LoadPal**.

3. The **Files** menu pops up. You will select an image file on which to base the 256-color palette. Since the first image we use in the sequence is **TUTOR1.PCX**, highlight it and press **Select**.
4. The **Wait** menu pops up. From the menu, select **Don't Wait**.
5. The **Next** menu pops up. Press **Select** to accept the default.
6. The Line Counter advances to Line 3.

*Lines 3 and 4 bring in the graphic images to set the stage for the Title events.*

### **Recreating Line 3**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Hrz\_In**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The image appears on the screen; the Line Counter advances to Line 4.

### **Recreating Line 4**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR2.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select **Tp\_Bot**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK**.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. Buzz wipes onto the background; the Line Counter advances to Line 5.

*Lines 5 and 6 demonstrate how to create the titles and place them on the screen.*

### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you would like to use.
3. The **SP/DP** menu pops up. Select **Permanent**. Select **OK**.

4. **Parameters** pops up. In the **Input Title** box, type **Hello!** Press **Select**. For **Choose Color from Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select a color for the title.
5. Text appears on the screen for positioning. Move the text to where you want it on the screen and press **Select**.
6. The **Wait** menu pops up. Select **Wait for Timeout**. Select on the **Seconds** box, type in **1** and press **Select**. Press **Select** again to accept 18th Seconds default.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 6.

### ***Recreating Line 6.***

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
3. The **SP/DP** menu pops up. Select **Permanent**. Select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **I'm Buzz**. Press **Select**. For **Choose Color from Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select a color for the title.
5. Text appears on the screen for positioning. Move the text where you want it on the screen and press **Select**.
6. The **Wait** menu pops up. Select **Wait for Timeout**. Click on the **Seconds** box, type in **2**, press **Select**. Press **Select** to accept 18th Seconds default.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 7.

*Lines 7 - 9 show you how to string Title events together to make a short statement on the screen.*

### ***Recreating Line 7***

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface.
3. The **SP/DP** menu pops up. Select **Permanent**. Select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **Hope you**. Press **Select**. For **Choose Color from Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select a color for the title.
5. Text appears on the screen for positioning. Move the text where you want it on the screen and press **Select**.

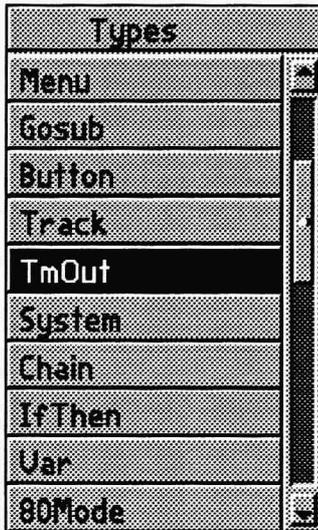
6. The **Wait** menu pops up. Select **Wait for Timeout**. Select on the **Seconds** box, type in **3** and press **Select**. Press **Select** again to accept 18th Seconds default.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 8.

### ***Recreating Line 8***

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select the same typeface you chose for Line 7.
3. The **SP/DP** menu pops up. Select **Permanent**. Select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **enjoyed**. Press **Select**. For **Choose Color from Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select the same color you selected in Line 7.
5. Text appears on the screen for positioning. Move the text to line up with the text from Line 7 and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 9.

### ***Recreating Line 9***

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select the same typeface you chose for Lines 7 and 8.
3. The **SP/DP** menu pops up. Select **Permanent**. Select **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **our show**. Press **Select**. For **Choose Color from Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select the same color you selected in Lines 7 and 8.
5. Text appears on the screen for positioning. Move the text to line up with the text from Lines 7 and 8 and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on the screen; the Line Counter advances to Line 9.
9. Rewind the sequence to Line 1 and press **F6** to see it play.



## Chapter 31

# A TmOut Event

The Timeout event is used to give the viewer an amount of time (programmable by you) to make a key press, menu selection, user window choice, or keyboard answer entry. Other timeouts refer to the system clock and cause events to occur as a result of elapsed time and system time. If the timeout occurs, a branch or subroutine you predetermine will automatically happen.

---

*NOTE: Once a timeout is set, it stays on and in effect until it is changed with another TmOut event, or turned off with the TmOut "Off" event.*

---

## Type

The field cursor is in the event Type field, and the event menu pops up. Select TmOut. Your choice appears in the event field.

## Method

Timeout
NoKeyJMP
NoKeySUB
KeyJMP
KeySUB
TimedJMP
TimedSUB
ClockJMP
ClockSUB
Off

The Timeout method lets you determine what kind of timing you will use. In all cases you have two options, jump on a timing, or Gosub on a timing. A jump branches to the specified label in the Next field after the timeout occurs. A Gosub goes to a subroutine after a timeout occurs. After a <ret> is encountered in the Next field of the subroutine, control returns to the line following the line where the timeout occurred. “Off” turns the timeout off. Only one timeout method can be set on at any one time.

### NoKeyJMP

This method is used in cases where you want the sequence to automatically branch if no key is pressed. The branch occurs when no key has been pressed for the amount of time specified in the Wait field, and the branch target is the label of the event entered in the Next field. This feature is generally used to restart a sequence automatically if no one has used it for a period of time.

### NoKeySUB

This method is used in cases where you want the sequence to automatically display a subroutine if no key is pressed. The Gosub occurs when no key has been pressed for the amount of time specified in the Wait field, and the branch target is the label of the event entered in the Next field. This feature is generally used to warn the user that the system restarts a sequence automatically if the user does not interact with the system for a period of time.

### KeyJMP

This method is used in cases where you want the sequence to automatically branch when a key is pressed. The branch occurs when a key has been pressed; otherwise linear processing continues. The branch target is the Label of the event entered in the Next field. This feature is generally used

when you are making an introductory loop in the start of a display and you want the loop to continue until someone presses a key.

### ***KeySUB***

This method is used in cases where you want the sequence to automatically display a subroutine when a key is pressed. The Gosub occurs when a key has been pressed. If a key is not pressed for the amount of time specified in the Wait field the linear execution of event continues. When a key is pressed, the Gosub occurs and the target is the Label of the event entered in the Next field.

### ***TimedJMP***

The timed jump method is used in cases where you want the sequence to branch automatically after a specified period of time. The branch occurs when the specified period of time has been reached; otherwise regular processing continues. The time is specified in the Wait field in hours:minutes:seconds. The counting of passage of time starts as soon as this TmOut event occurs. The branch target is the Label of the event entered in the Next field.

### ***TimedSUB***

The timed jump method is used in cases where you want the sequence to Gosub automatically after a specified period of time. The Gosub occurs when the specified period of time has been reached; otherwise, regular processing continues. The time is specified in the Wait For field in hours:minutes:seconds. The counting of passage of time starts as soon as this TmOut event occurs. The Gosub target is the Label of the event entered in the Next field.

### ***ClockJMP***

The clock jump method is used in cases where you want the sequence to branch automatically at a specified moment in time. The branch occurs at the specified time according to the computer system clock; otherwise, regular processing continues. The time is specified in the Wait For field in hours:minutes:seconds. The branch target is the Label of the event entered

in the Next field. This is used in sequences that require attention to the time of day. Be sure that your system clock is set properly using 24-hour time.

### ***ClockSUB***

The clock Gosub method is used in cases where you want the sequence to Gosub automatically at a specified time. The Gosub occurs when the specified time has been reached; otherwise, regular processing continues. The time is specified in the Wait field in hours:minutes:seconds, set as 24-hour time. The Gosub target is the Label of the event entered in the Next field.

### ***Off***

This method is used to turn the current TmOut method off. Remember that only one TmOut method is active at any time.

### ***Wait***

---

This entry sets timing information for the timeout. With the “Key” and “No Key” timeouts, the time is set in seconds and 1/18’s of seconds. The system accepts a touch screen touch, mouse movement, a keypress, serial I/O, or an AskLabel as input. In the timed TmOut events, the Wait is set in hours:minutes:seconds. These are time-based only, and are reset with a new TmOut event or by turning the TmOut off.

### ***Other Fields***

---

Label, File, Sp/Dp, Parameters, and Next are fields common to most lines. Apply them as necessary.

### ***Chapter Example***

---

This example tells TEMPra Media Author to jump to a line labeled “any” at 1:30 PM (13:30).

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	TmOut		ClockJMP			13:30:00	any



## Chapter 32

# A Track Event

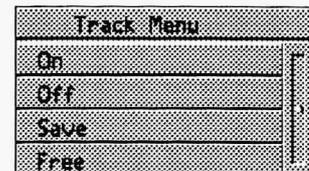
Keep track of what your audience is doing. Save this to a file for a permanent record or use it for internal branching to customize the presentation for the viewer. The track saves all variable values, the most recent choice from any labeled menu or labeled user window. The information stored is the number of the choice, in the list as it appears in the Workform and the Label of the event.

## Type

The Track type allows you to monitor and save all of the responses of every viewer who interacts with the TEMPRA Media Author system. Select Track from the Type menu.

## Method

You can turn the tracking on or off, and you can save the track in a file. Each viewer's record is appended to the file. The name of the file is the same as the name of the sequence, but the extension is TRK. The track is saved in the same directory where TEMPRA Media Author resides.



## Track Event

---

***NOTE:** After a track is saved, the internal memory of the track is cleared.*

### **Other Fields**

---

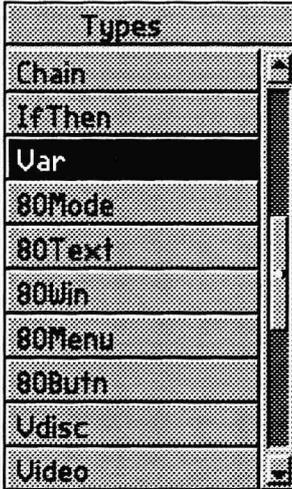
Label, File, Sp/Dp, Parameters, and Next are fields common to most lines. Apply them as necessary.

### **Chapter Example**

---

This example turns answer tracking on.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Track		On				



## Chapter 33

# A Var Event

Unlock real interactive power with variables. The Variable event is used to declare variables that can then be modified, displayed, or entered from the keyboard. They can be read from and written to text files. Variables can be numbers or character strings. Once the variables have values, they can be used in IfThen events, saved to or loaded from files. Var files can be given names from string variables. Variables can also be used in the Type, File, Method, Sp/Dp, Parameters, or Wait fields. By assigning values to the variables, you can get one line to do just about anything.

All variables, or **Vars**, must first be declared before they can be modified, tested, or displayed. Variables can be string, noted with a "\$"; numeric, noted with a "#"; or real time for date and time, noted with an "&." Real-time variables do not have to be declared because TEMPRA Media Author does that for you. Month is "&month," day of month is "&mday," day of week is "&wday," year is "&year," hour is "&hour," minute is "&min," and second is "&sec."

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$word = "Media Author"		

This example creates a Var named "\$word" that contains the words "Media Author."

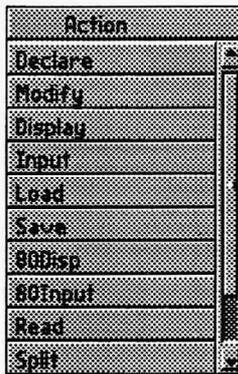
## Type

---

Variables are used for many purposes: Scorekeeping, giving weights to different answers, creating simulations, naming files, allowing viewers to enter answers from the keyboard, and so on. You will develop your own uses for them. Variables **must** be declared before modification, display, loading, or saving.

## Method

---



The Method field is used to tell the TEMPRA Media Author system what you are going to do with a Var.

### **Declare**

State the name, kind of variable, and give it an initial value.

### **Modify**

Change the value of the Var.

### **Display**

Write the value of the Var on the video screen in any of the fonts and in any color available in the palette. This field holds the name of the font.

### **Input**

Receive keyboard input and place the entry into the Var.

---

**NOTE:** When you choose to **Display** or **Input** a Var, the Method field contains the name of the font with which you choose to display the Var.

---

**Load**

Load a file of variables in from a disk file.

**Save**

Save variables to a disk file. Save the Vars that match the series, or all Vars with “\*.”

**80Disp**

Write the value of the Var on the video screen in any 80 column text mode of 16 colors.

**80Input**

Receive keyboard input and place the entry into the Var while in 80 column text mode.

**Read**

Read a line in from a text file and place the string into a string variable.

**Split**

Divide a string Var into a series of Vars with similar names. To do this, you need to use a separator.

**Write**

Write the values in the chosen Vars to a text file.

**Sp/Dp**

---

Speed does not apply, but you do need to choose a display.

**Display**

Display is only applicable to the Display and Input Var method. In this case, you can choose to make the display of your Vars Permanent, Temporary, or Erasing after Display.

## Parameters

The Parameters field is where the kind of variable (numeric or string) is determined, the name is stated, the values are set or changed, and the display information is placed. The parameter choices you have depend on the method you are using. With Vars, “#” means a numeric Var, “\$” means a string Var, and “&” means an internal system Var. When using Vars, the name of the Var represents the current value that Var has been set to.

### Declare

When you declare a Var, you make it either a string Var, where the Var contains words, or a numeric Var, where the Var contains numbers. String variables can be used for real number activities when you use “N” operators.

You create a Var name, then you give it an initial value. “ ” or space is a traditional initial value for strings, and 0 is the traditional initial value for numeric vars.



Sharp Boy says:  
Numeric variables are integers, so they are only whole numbers.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$word = ""		
	Var		Declare		#num = 0		

### Modify Numeric Vars

Modifying a Var is simply changing the value stored in the Var. The modifications you make are done by operators. These operators determine the kind of modifications that are made to the variables. The maximum values for numeric Vars is 32767, and the minimum value is -32768. This is because numeric Vars are integers.

The operators that modify numeric Vars and what they do are:

- = This makes the left Var equal to the contents of the right Var or entered number.
- += This makes the left Var equal to the sum of its current value and the value of the right Var or entered number.

- = This makes the left Var equal to the difference of its current value and the value of the right Var or entered number.
- \*= This makes the left Var equal to the value of the right Var or entered number times the current value of the left Var.
- /= This makes the left Var equal to the value of the left Var divided by the right Var or entered number.
- %= Results in the left Var being equal to the value of the remainder of the division of the left Var by the right Var or entered number.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Modify		#num2 - 456		
	Var		Modify		#num = 123		
	Var		Modify		#num += #num2		
	Var		Modify		#num *= 10		

The result of the above modifications is “#num = 5790, or (123 + 456) x 10.”

### ***Modify String Vars***

Modifying a Var is simply changing the value stored in the Var. The modifications you make are done by operators. These operators determine the kind of modifications that are made to the variables.

The operators that modify string Vars and what they do are:

- = This makes the left Var equal to the contents of the right Var or entered characters.
- += This makes the left Var equal to the current value of the Left Var with the contents of the Right Var added to it.

You handle strings of numerals as real numbers when you use the following:

- N+= This makes the left Var equal to the sum of its current value and the value of the right Var or entered number.

- N-=** This makes the left Var equal to the difference of its current value and the value of the right Var or entered number.
- N\*=** This makes the left Var equal to the value of the right Var or entered number multiplied by the current value of the left Var.
- N/=** This makes the left Var equal to the value of the left Var divided by the right Var or entered number.

### ***Display Numeric and String Vars***

The Display method parameters are used to identify what Var is to be displayed, where it is to be displayed (x,y coordinates), and in what color it is to be shown. The only difference between showing numeric and string Vars is the type of Var you choose to show.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Modify		\$word2 = "number 2"		
	Var		Modify		\$word = "Example "		
	Var		Modify		\$word += \$word2		

The result of this will be \$word = "Example number 2."

When you edit this Parameters field, a menu gives you these choices.

Var \$word2 will be shown at 10x and 250y in color number 2.

Var #num will be shown at 120x and 75y in color number 14.

Color and positioning for Vars is the same as it is for titles. Please refer to that section for more details.

### ***Input for String Vars***

The Input method allows users to enter information from the keyboard. The parameters field is used to identify the Var that is to hold the entered information to tell TEMPRA Media Author how many characters to allow, what kind of input to allow, and where the entry will be seen on the screen. The font used to display the input is always the default font that is set with the Global event.

Set?
All
Variable
Position
Color

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Roman1		\$word2 @10x 250y 2c		
	Var		Times25		#num @120x 75y 14c		

**Name the Var** that holds the entry.

**Set the length** of the string. Any length from 1 to 256 characters is fine. “Keystroke Only” input type has to be 1 character in length.

**Choose the input type.** Since this information can be anything from a single keystroke to a sentence that is 256 characters long, TEMPRA Media Author has to know what kind of input to expect.

Your choices of input types are:

**1** - Alphanumeric = a...z, A...Z, 0...9

**2** - Numeric = 0...9

**3** - Printable= all of above and symbols

**4** - Alphanumeric or Space = a...z, A...Z, 0...9, ‘ ’

**5** - Filename = any legal filename characters

**6** - Keystroke string = keyboard return code\* (length is always 1)

\*See Keyboard Return Codes, Appendix F, in this manual.

**Set the Location** for the input to be displayed, (X,Y coordinates)

### ***Input for Numeric Vars***

The Input method allows users to enter numeric information from the keyboard. The parameters field is used to identify the Var that is to hold the entered information. It tells TEMPRA Media Author how many characters to allow, and where the entry will be seen on the screen.

**Name the Var** that holds the entry.

**Set the length** of the string. Any length from 1 to 5 characters is fine. The maximum integer value is 32767, and the minimum is -32678.

The input type is automatically set to 2 for numeric.

Set the Location for the input to occur, (X,Y coordinates).

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Input		\$word2 1T 5L @50x 25y 7c		
	Var		Input		#num 2T 4L @120x 75y 14c		

The first example creates input to string Var \$word2, with input type 1 (alphanumeric), of length 5, at the x,y coordinates, and shown on the screen in color 7. The second example shows input to numeric Var #num, of type 2 (numeric), of length 4, at the x,y position, in color 14.

### Load

In the parameters field, you specify the Var that holds the name of your file. When this event line is executed, all the Vars in your file are loaded into TEMPRA Media Author.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$file = "varfile.var"		
	Var		Load		\$file		

### Save

In the Parameters field, you specify the Var that holds the name of your file. Then you specify the series of Vars you want to save. You can save all the Vars by entering the "\*" character without any prefix. If you have made a series of Vars that all start with the same three characters, for example, \$VG1, \$VG2, \$VG3...TO \$VG25, you could save only these Vars with the following event lines.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$file = "varfile.var"		
	Var		Save		\$file \$VG*		

If you wanted to save all the Vars, you would use the following.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$file = "varfile.var"		
	Var		Save		\$file *		

### ***80Disp***

This is very similar to the Display method, except for the fact that this displays the values of the Vars in the 80 column text mode. There are some key differences, notably the color is not specified, and the font is always the same.

The 80Disp method parameters are used to identify what Var is to be displayed, and where it is to be displayed (x,y coordinates). The only difference between showing numeric and string Vars is the type of Var you choose to show.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		80Disp		\$word2 @10x 250y		
	Var		80Disp		#num @120x 75y		

Var \$word will be shown at 10x and 250y.

Var #num will be shown at 120x and 75y.

The positioning is done the same way here as it is done in the 80Text Parameters event. Please refer to that section for more details.

### ***80Input for String Vars***

The 80Input method allows users to enter information from the keyboard while the system is in 80 column text mode. The Parameters field is used to identify the Var that is to hold the entered information, which tells TEMPRA Media Author how many characters to allow, and where the entry will be seen on the screen.

**Name the Var** that holds the entry.

**Set the length** of the string. Any length from 1 to 256 characters is fine. “Keystroke Only” input type has to be 1 character in length.

**Choose the input type.** Since this information can be anything from a single keystroke to a sentence that is 256 characters long, TEMPRA Media Author has to know what kind of input to expect.

Your choices of input types are:

1 - Alphanumeric = a...z, A...Z, 0...9

2 - Numeric = 0...9

3 - Printable= all of above and symbols

4 - Alphanumeric or Space = a...z, A...Z, 0...9, ‘ ’

5 - Filename = any legal filename characters

6 - Keystroke string = keyboard return code\* (length is always 1)

\*See Keyboard Return Codes, Appendix F, in this manual.

**Set the location** for the input to occur, (X,Y coordinates).

### ***80Input for Numeric Vars***

The Input method allows users to enter numeric information from the keyboard. The Parameters field is used to identify the Var that is to hold the entered information, which tells TEMPRA Media Author how many characters to allow and where the entry will be seen on the screen.

**Name the Var** that holds the entry.

**Set the length** of the string. Any length from 1 to 5 characters is fine. The maximum integer value is 32767, and the minimum is -32678.

**The input type** is automatically set to 2 for numeric.

**Set the Location** for the input to occur, (x,y coordinates).

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		80Input		\$word2 1T 5L @50x 25y		
	Var		Input		#num 2T 4L @120x 75y		

The first example creates input to string Var \$word2, with input type 1 (alphanumeric), of length 5, at the x,y coordinates. The second example shows input to numeric Var #num, of type 2 (numeric), of length 5, at x,y coordinates.

### **Read**

The Read method is used to read lines from ASCII text files so they can be used in TEMPra Media Author. The Parameters field identifies the string Var that holds the filename to be read in, defines which line is read, and which string Var the information is stored in. Remember, if the value is “^Z” then you have reached the end of the file.

**Name the Var** that holds the filename.

**Set the line number** to read.

**Choose the string Var** that holds the new information.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$file = "Demo.txt"		
	Var		Declare		\$line2 = ""		
	Var		Read		\$file 2 \$line2		

This example creates two new Vars, \$file which holds the file name, and \$line which holds the read in information. The third line reads in line number 2 into Var \$line2.

### **Split**

The Split method is used to split up long lines that are read in to string Vars with the Read method. There are also general uses for Split in splitting user input into individual string Vars that can then be analyzed individually. The Parameters field is where the long string Var line is identified. The delimiter characters are defined, and the string Vars that are in a series are named.

**Name the Var** that holds the long string to be split.

**Name the Var** that holds delimiting characters.

**Set up the series of string Vars** that are to hold the individual elements of the long string.

For this example we use the Vars from the above example. The content of Var \$line2 is:

**“The |quick |red |fox |sure |got |a |jump |on |that |dog.”**

Notice the “|” character between all the words. It is the delimiter and has to be in the string that is read in. It can be any character, or series of characters.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$delim = " "		
	Var		Split		\$line2 \$delim \$VG		

The result will be 11 string Vars, \$VG1, \$VG2, \$VG3 ... \$VG11, each with a separate word from the above line. \$VG1 = “The”, \$VG2 = “quick”, \$VG3 = “red”, and so on. Notice that it is not necessary to declare each of the series of \$VG Vars, TEMPRA Media Author does this for you. You will notice that the delimiting characters are not part of the series, and the last string Var \$VG11 has “ ” as a value.

### **Write**

The Write method is used to write lines to ASCII text files. The Parameters field identifies the string Var that holds the file name to be written to, and defines which Vars are to be written, if you are writing a series of Vars. The Parameters field allows you to specify if you are going to overwrite any existing file that has the same name, or if you are going to append new lines to the existing file.

**Name the Var** that holds the file name.

**Identify the Var series**, if any, by use of a three character series prefix. If all Vars are to be written, select “\*” without a series prefix.

**Choose to Overwrite or Append.**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Declare		\$file = "New.txt"		
	Var		Write		\$file \$VG* A		

This example creates a new Vars \$file that holds the file name "New.txt." The second line writes all string Vars that start with "VG" to the file and appends it to the end.

If instead of using \$VG\*, we had used VG\*, then all Vars, both string and numeric, would be written to the file "New.txt".

---

**NOTE:** You cannot assign a value to an internal ("&") Var. These Vars get their value from the system.

---



---

**NOTE:** When modifying strings: \$Demo1 = "First," \$Demo2 = "Second," \$Demo1 + \$Demo2 creates \$Demo1 = "First Second."

---

**Return codes** come back as strings, so set the IfThen tests for "code." See Keyboard Return Codes in Appendix F of this manual.

## Next

---

After your choice is entered for Wait, your cursor moves to Next. Notice that you see a requester window. This means that Next is a field where you type something rather than select an option. Type the label of the next event line you want to execute or simply press [Enter] to accept the default, which goes to the next line.

## Var—Extended Capability

---

To provide the most flexible and powerful system available today, Mathematica, Inc. is now providing a quantum leap in authoring software. It is now possible to put variables in any field on the workform except the Label or Next fields. These variables are interpreted as event types when they are in the Type field, as filenames in the File field, Methods in the method field

and so on. This also means that the x,y positions in the Parameters field can be changed while the sequence is running, just by modifying the values of the x and y variables. This allows you to change the content of an event line just by reassigning variables.

In order to use this successfully, you must declare the variables, then assign values to them that are understandable by TEMPRA Media Author. For example, if you have a Var declared as \$type initially set to “ ” then for that var to be used you have to give it a value that is the same as the event types TEMPRA Media Author already uses. For example \$temp=“Screen” works if you then manually enter the \$temp variable name into the Type column on the workflow. If you are making the parameters in the Parameter field variables, you have to have one variable for each parameter. The maximum number that is meaningful to TEMPRA Media Author is six different parameters in the Parameters field.

### ***Var—Reserved Variable Names***

There is a host of reserved variable names. These reserved names allow the designer to access system information, allow the system to pass information to the designer as a result of certain events, and provide the designer with access to files as they are available through the \*.PTH file for the sequence.

<b><u>NAME</u></b>	<b><u>MEANING</u></b>	
<b>Internal Variables</b>		
&month	Month	1..12
&mday	Day of month	1..31
&wday	Day of week	1..7
&year	Year	
&hour	Hour	
&date	Complete date	
&min	Minute	
&sec	Second	
&12time	Current 12 hour time	
&24time	Current 24 hour time	
&hres	Horizontal Resolution	
&vres	Vertical Resolution	

<b><u>NAME</u></b>	<b><u>MEANING</u></b>
<b>&amp;outmode</b>	Current Output Mode
<b>&amp;program</b>	Program Directory (string value only; do not assign to numeric.)
<b>&amp;screen</b>	Screen Directory (string value only; do not assign to numeric.)
<b>&amp;sprite</b>	Sprite Directory (string value only; do not assign to numeric.)
<b>&amp;text</b>	Text Directory (string value only; do not assign to numeric.)
<b>&amp;sequence</b>	Sequence Directory (string value only; do not assign to numeric.)
<b>&amp;audio</b>	Audio Directory (string value only; do not assign to numeric.)
<b>&amp;font</b>	Font Directory (string value only; do not assign to numeric.)
<b>&amp;font</b>	Variables Directory (string value do not assign to numeric.)
<b>&amp;font</b>	DBMS Directory (string value only; do not assign to numeric.)

### **Numeric Variables**

<b>#_com1</b>	Com port value from port 1
<b>#_com2</b>	Com port value from port 2
<b>#_com3</b>	Com port value from port 3
<b>#_com4</b>	Com port value from port 4
<b>#_posx</b>	Returned X value from a selection made during Button event
<b>#_posy</b>	Returned Y value from a selection made during a Button event

### **String Variables**

<b>\$_file</b>	Returned file name from a Dir event
<b>\$_line</b>	Returns the line from text files.
<b>\$_word</b>	Will return the word in a line number from text files.

### **DBMS Variables**

Please check the DBMS section for listing and details.

**General Note about Reserved variable names:** Mathematica, Inc. may be expanding this list of reserved variable names. If you want to be safe for now and in the future, do not use the “&” or “\_” characters as the first character in any of your variable names.

## ***Using Variables Successfully***

Variables are, of course, things that change. This makes them both powerful tools for the careful and the source of great confusion to the uninitiated. The basic approach to take with Vars is to declare them, then modify them. After that, you can save them to files. In addition they can be used in the IfThen event to provide the basis for an internal branch. For example, if Var1 is greater than 4, then branch to right.

## ***Using Vars to get to Files***

In TEMPRA Media Author, files are accessed through string Vars. The file name is placed into a string Var, then that var is used as the holder of the name.

Once a Var is declared, you can give it a value. When you are ready to record the values of the variables, you can save them to a variable file. The variable files are ASCII files with the following format.

```
$Var1 = "contents"  
$Var2 = "Any value"  
#Num1 = 123  
#Num2 = 456
```

Once a Var is declared, you can load a value for that Var in from a file. When you do this, the value of that Var is immediately changed to whatever came in from the file.

## ***Reading and Writing with Text Files***

Once a string Var is declared, you can read a line into that var from any text file. The next step is to split the line up into individual Vars that can be analyzed. The opposite of split is modify with the "=+" symbol. This action adds the values in the variables together. Once the string Vars are constructed, they can be written to a file. When writing to a file, the contents of the Var can be added (appended) to a file, or written over the current contents of the file.

---

## ***Series of Variables***

It is a good idea to have the ability to load and save only some Vars and not all of them. How do you control which ones are saved? TEMPRA Media Author allows you to create a series of Vars that you can save and write, using a wild card character “\*.”

To do this, create Vars with names that all start with the same two characters, for example, \$VR1, \$VR2, \$VR3, \$VR4 and so on. When you save them, you can do this by specifying to save all VR\* Vars. TEMPRA Media Author reads the Var list and saves only the ones with the first two characters equal to VR.

When you use the Split method, you create a series of Vars with similar names.

## ***Seeing the Values of Variables***

You can press the [Alt] V key combination to get a list of the current Vars and their values. To assign new values to them, use the “Preview by line” feature of TEMPRA Media Author. Press [F5] on the Var lines you want to use to modify the contents of the current Var list.

## ***Using the Read Method For Vars***

First you create a string Var to hold the filename, if you do not already have one. Then you modify that string Var and make it equal to the name of the file you want to read. Do this when you see:

**Input Var Name Containing Filename - Name : \$ \_\_\_\_\_**

Next, tell TEMPRA Media Author which line in the file to read:

**Line Number to Read - Line : \_\_\_\_\_**

Now you have to tell TEMPRA Media Author which string Var you want to hold the information you are reading in from this line in the chosen file:

**Input Var Name for Line - Name : \$\_\_\_\_\_**

If the line contains the symbol “^Z”, you know you have reached the end of the file.

## ***Using the Split Method for Vars\_\_\_\_\_***

When you have read a line from a text file into a string Var, or you have a string Var that contains several words, you can Split the string Var up into a series of string Vars. First you select the string Var to split:

**Input Var to Split - Name : \$\_\_\_\_\_**

Then, you have to tell TEMPRA Media Author how to tell the difference between one entry in the long string and another entry. TEMPRA Media Author looks for delimiter characters. These can be spaces or any characters you can type, such as “[”, “}”, “/”, and so on. To use this delimiter you have to create a string Var to hold the delimiter. Then you have to place the delimiter into the string Var. Now you can use that string Var in this split when prompted:

**Input Var Containing Separators - Name : \$\_\_\_\_\_**

Since there is no way for you as an author to know for sure how many elements there are going to be in the strings you split, TEMPRA Media Author generates the series of Vars for you, using a prefix of your choosing. You do not have to pre-declare these Vars, you only have to choose a prefix that can be used. When the split is executed, a series of Vars are generated numbering from 1 to the number of elements in the string. Enter the prefix when you see:

**Input Var Prefix - Name : \$\_\_\_\_\_**

---

## ***Using the Write Method for Vars***\_\_\_\_\_

First, you create a string Var to hold the file name, if you do not already have one. Then you modify that string Var and make it equal to the name of the file you want to write. Do this when you see:

**Input Var Name Containing Filename - Name : \$**\_\_\_\_\_

If you are writing a series of Vars to a text file, you can identify the series by using a name pattern and the wild card “\*.” If you want to write all the variables to a text file, use the wild card “\*” without any name patterns. The name patterns are generated by splitting with a prefix, or making your own, for example; \$VG1, \$VG2, \$VG3, \$VG4.

**Input Var Name Pattern - (\* matches all) : \***\_\_\_\_\_

You are prompted to choose Append (to add the current variable values you are writing to a file) or Overwrite (overwrite an existing file with the current values).



## Chapter 34

# A VCR Event

This event allows you to control any SONY VISCA video source from the computer. Using a SONY Vbox, you can control video sources ranging from VCRs to camcorders. Or you can use a SONY Vdeck for Hi-8 videotapes. You can view video coming into the computer by setting a video window (see VidWind in the Video chapter) or linking to an RGB monitor (your television set will work if you have video and audio input/output).

TEMPRA Media Author allows you to control up to seven video sources from the computer, giving you true video production capabilities. In fact, with TEMPRA Media Author, if you combine your computer with two monitors and several VCRs, you can turn your work area into a full-fledged editing suite! If you are using any SONY VISCA video devices, consult the hardware documentation for proper hookup.

Before creating a VCR event using VISCA control, be sure to change your CONFIGMA to reflect VISCA support. Three selections to pay particular attention to are: Video Input Standard, PC-VCR Type, and PC-VCR Port. See the Technical Information Appendix on using the TEMPRA Media Author Configuration Program for details.

## Type

---

Select VCR from the Type menu.

## File

---

In TEMPRA Media Author, a menu pops up displaying seven decks to choose from. Choose the deck you want to control from the computer. If a message appears saying "0 VISCA DECKS FOUND," follow the Vdeck or Vbox manual to ensure correct cable attachment. Also, be sure that the VISCA deck protocol is selected in the CONFIGMA. To verify that VISCA is enabled, start TEMPRA Media Author by typing MA -V. You can also tell if VISCA is enabled if the amber VISCA light is lit on your Vdeck.

## Method

---

Method
Play
Stop
Rewind
Search
SearchBH
Pause
Playseg
Record
Zero_HMS
InaEdit1

Once TEMPRA Media Author has "found" a VISCA deck, it assumes control of the deck functions. Normal modes of operation directly from the Vdeck are now controlled by the computer and TEMPRA Media Author.

### **Play**

This sets the tape in motion for standard viewing. Wherever the tape was last stopped is where it will resume. Parameters will not apply to this method.

### **Stop**

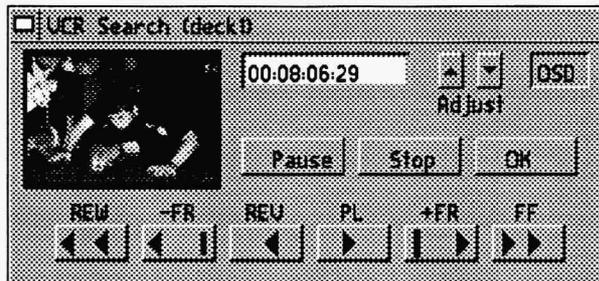
Use this command to stop normal tape operation.

Parameters will not apply.

### **Rewind**

Use this to rewind the tape. Parameters will not apply.

## Search



Use this to find specific segments of footage to be shown. Once you have found the section, be sure to note its beginning (the numbers of the time stamp in the menu window). You'll need

to know these times if you intend to use a Play Seg type line.

If your VCR reads Sony time code, be sure you are using videotapes that have a time code recorded throughout the entire length of the tape, even if you have only recorded a few minutes of video on a two-hour tape. If you do not, and the VISCA device plays beyond your time code, the device will automatically fast-forward in search of the starting point of the time code. This could take to the end of the tape with little or no time to completely rewind during the execution of a sequence.

Once you've set TEMPRA Media Author to search, you can quickly call up the search mode by selecting in the Parameters field. The Search menu pops up on the screen. When you designate your search point, press "OK" on the VCR Search menu. TEMPRA Media Author automatically searches for that point when you execute the line.

## SearchBK

This method performs a search similar to the Search event, but the search occurs while other events execute.

## Pause

This pauses the tape. Parameters do not apply.

## Play Segment

This method allows you to play a very precise amount of video exactly where you want it in your sequence. Used in conjunction with search methods, you

can use several segments of one tape. The procedure would be to search to a certain point and then play a segment.

### ***Record***

This will record onto the tape any video signal coming into the Vdeck's video input ports or Vbox. Using TEMPRA Media Author and any video board with video out capability (such as the Cardinal SNAPplus board), you can record your sequences to a Hi-8 master tape for mass duplication and distribution.

### ***Zero\_HMS***

This function returns your hours-minutes-seconds counter to zero when you don't have Sony RC time code.

### ***InsEdit1***

This method is used to select a segment of video to be recorded onto the record deck. Be sure the record deck is at the right point in the tape for this insert edit.

## ***Parameters***

---

The Parameters field displays such information as the search points and the Start and Stop points if you are editing with TEMPRA Media Author.

### ***Search***

A VISCA controller menu appears. The buttons on this menu will control the VISCA to search the tape for a specific place or segment. To prevent delays in your sequence, tell TEMPRA Media Author and the VISCA to search for a certain place on the tape well in advance of the time when you intend to play it. TEMPRA Media Author will not go forward until the search is complete.

### ***SearchBK***

A VISCA controller menu appears. The buttons on this menu will control the VISCA to search the tape for a specific place or segment. To prevent delays in your sequence, tell TEMPRA Media Author and the VISCA to search for a certain place on the tape well in advance of the time when you intend to play it. You can show graphics, images, and animations while TEMPRA Media Author is searching for the point on the video you want to begin playing.

## ***Play Segment***

A VISCA controller menu appears for you to search and find the starting point for the segment to be played. Find the starting place and press [Enter]. The next menu to appear asks you to find the ending point of the video segment. Find this point and press [Enter].

During the sequence, TEMPRA Media Author automatically searches for this segment once the line is executed. You may want to use a search line several lines prior to a play segment to avoid a delay in your sequence.

## ***InsEdit1***

A VISCA controller menu appears for you to search and find the starting point for the segment to be played. Find the starting place and press [Enter]. The next menu to appear asks you to find the ending point of the video segment. Find this point and press [Enter].

## ***Wait***

For the most part, Wait is like any other line. If you are going to play a segment, you must add a Time Out period so TEMPRA Media Author will not go on to the next line and bypass playing the segment. The VISCA will play, but you will not see anything. When this event is executed, the video segment set forth in the Parameters field will be recorded out to the record deck at the point where the tape sits.

## ***Other Fields***

Label, Next, and Wait are fields common to most lines. Apply them as necessary.

## ***Video Editing***

You can use TEMPRA Media Author to record to videotape, editing from various SONY VISCA sources. You can perform straight deck-to-deck editing and recording in TEMPRA Media Author, using the computer to set the edit points. Or you can set up a presentation in TEMPRA Media Author and record it to videotape using the video window capability. Use the

Workform to create a video edit sequence, which you can modify quickly, changing your start and stop edit points on the fly.

For best results when editing, we suggest you use an RGB monitor (one monitor for each video source is ideal, so you can watch what is coming into and going out of your computer). The only video you will see in the computer (whether in your VCR Search window or in a video window) is the video connected to your video board. So, although you may have seven SONY video devices daisy-chained together, you will only see video in the computer from the video device that is actually connected to your computer's video board input port.

The following example might give you an idea of how to edit videos in TEMPRA Media Author.

### Example of deck-to-deck editing, Deck 1 is target deck:

Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	VCR	Deck01	Search		0:00:30:00		
	VCR	Deck02	InsEdit1		0:00:10:13 0:00:20:26		
	VCR	Deck03	InsEdit1		0:00:30:09 0:00:56:19		
	VCR	Deck03	InsEdit1		0:03:15:01 0:03:17:02		
	VCR	Deck02	InsEdit1		0:01:23:05 0:01:24:01		

**An example of using video in a presentation, incorporating graphics, text, animation, and audio:**

Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
Top	VCR	Deck01	Search		0:03:22:05		
	VCR	Deck01	Pause				
	BColor		Dissolve		Solid 16c		
	Text	INTRO	Helv30	E 003	Ox 282y 153c 45w 4h nnm		
	VCR	Deck01	Play				
	Video		SetMode		640x480x256		
	Video		VidWind	P	0bx 0by 639ex 479ey		
	Video		VidTcol		0c		
	Video	CREDIT	Scrollup	1E	125bx 300by 1c		
	Screen	BUZZ1	Bot_Tp	99	163bx 125by 476ex 377ey		
	Video		VidNorm				
	VCR	Deck01	Stop				Top



## *Chapter 35*

# *A Video Event*

The Video event includes a number of selections that perform diverse tasks, from setting your display mode to showing animation and video. Because the Video event is one of the most comprehensive event types, this chapter is set up differently than the other event chapters. The chapter is divided into four sections: Video Display, Animation, Motion Video, and Infowindow. The Video event selections listed on the pop-up menu are not necessarily in the same order as this chapter, but remember you can sort the menus with [Alt] S. While these four sections are interrelated in some ways, they are also very different.

The Label, Sp/Dp, Parameters, Wait, and Next fields are common to most lines, but are not needed by all events in this chapter. Therefore, these fields will only be discussed when needed.

The **Video Display** section describes how to set the display mode, fade images in and out, load different color palettes, and set a viewport on the screen. These functions control the view you get on your computer monitor.

The **Animation** section focuses on bringing animation to a sequence from animation (FLC, FLI or FLX) files. Also included in this section is animation of text and colors on the screen.

The **Motion Video** section explains how you can play live or taped video in a window within your sequence. If you have a video input/capture board, like the Video Blaster or SNAPplus card, you can use live or taped video events from any video source. Also, you can output sequences to videotape.

The **Infowindow** section applies only to IBM Infowindow touch screen capabilities.

A chapter example sequence is available in the \TEMPRA\SEQ directory and is named VIDEO.SEQ. Several Video events are in this sequence to show you how they work. A copy of the Workform and a brief explanation of each line is at the end of this chapter.

## ***Video Display***

---

This section describes how to set the display mode, fade images in and out, load different color palettes, and set a viewport on the screen. These functions control the view you get on your computer monitor.

### ***LoadPal***

---

LoadPal allows you to load color palettes from the images you plan to use in your sequence. When you start TEMPRA Media Author, the automatic palette loading happens whenever you load an image. If you use LoadPal in advance of a Screen event, you can avoid unexpected palette shifts.

#### ***Type***

Select Video from the Type menu.

#### ***Method***

When you select LoadPal from the Video menu, you immediately go to the File field and the list of images pops up. The color palette you choose will be active until another LoadPal or ResetPal is selected from the Video menu.

#### ***File***

Select an image you plan to use in your sequence to set the color palette.

## **ResetPal**

---

ResetPal resets the palette to the TEMPRA Media Author default palette.

### **Type**

Select Video from the Type menu.

### **Method**

Select ResetPal from the Video menu.

## **SetMode**

---

TEMPRA Media Author automatically switches video modes to accommodate the images you are using when you load them in. This event allows you to choose the video mode in TEMPRA Media Author at any time during a sequence. The modes available to you depend on the abilities of your display adapter.

SetMode allows you to move from 320x200x256 color images, to 1024x786x256 color images, to 640x480x32,000 color images. When you change video modes, the screen will be cleared.

---

***NOTE:** You can set the default video mode by adding a statement to your AUTOEXEC.BAT file that causes TEMPRA Media Author to start in your selected mode. Any mode that operates with your display adapter may be used with the SET VVMODE= statement. For example, the statement SET VVMODE=VGA\_13 causes TEMPRA Media Author to start in 320 x 200 256 color mode.*

---

### **Type**

Select Video from the Type menu.

### **Method**

From the Video menu, select Set Mode. The field cursor goes to the Parameters field and a menu pops up with a list of Modes, Resolutions and Colors.

### ***Parameters***

Make your choice from the menu. Your choice is dependent on the memory to support the mode you select, the hardware to support it, and an image to display that is compatible with the mode. If you do not have EMS or HIMEM, you may get very slow display or other unexpected results from using video modes above the IBM standard VGA modes.

### ***SavePic***

---

This method allows you to save all or part of what you currently have on the VGA graphics screen. It goes to a file in the screen directory as a PCX file in low-color mode, or TGA in high-color mode.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select SavePic from the Video menu. To use Save Picture, enter a filename for a screen-captured image that is automatically saved as the sequence is run. You then write a screen event line using the filename of the screen capture from the Save Picture line. When the sequence is run, it saves whatever is on the screen and incorporates that fresh image directly back into the sequence at the point you place it in a Screen event. TEMPRA Media Author always executes the line you just made. This automatic execution causes TEMPRA Media Author to save the screen with the GUI.

### ***Parameters***

When you save part of an image, use the mouse to create a box around the area you want to save.

### ***FadeIN***

---

FadeIN allows you to fade an image in from black. (This feature may not work on all VGA cards.)

### ***Type***

Select Video from the Type menu.

## **Method**

Select FadeIN from the Video menu. The cursor goes to the File field.

## **File**

The File directory pops up. Select the image that has the palette you want to use. In order to use this feature, the image you want to fade in has to already be loaded in with a Screen event. This event only affects the palette of the image.

---

**NOTE:** *To fade an image in from a black screen, the following steps must precede the FadeIN event. These lines must be together as they are listed below. Of course, the filename can be changed. Set the speeds at the fastest possible, as FadeIN/FadeOUT take a bit of time to run.*

<i>Type</i>	<i>File</i>	<i>Method</i>	<i>Parameters</i>
<i>BColor</i>		<i>Quick</i>	<i>0c (black)</i>
<i>Screen</i>	<i>Slide20</i>	<i>Preload</i>	
<i>Video</i>	<i>Slide20</i>	<i>FadeOUT</i>	
<i>Global</i>		<i>AutoOff</i>	
<i>Screen</i>	<i>Slide20</i>	<i>Quick</i>	
<i>Video</i>	<i>Slide20</i>	<i>FadeIN</i>	

*The following sequence brings an image to the screen, fades it out, and then brings the color back to the screen with a BColor event and then another Screen event. If you have used a Global event, AutoOff, earlier in your sequence, you must do another Global event, AutoSet, before this sequence. The default is AutoSet.*

<i>BColor</i>		<i>Quick</i>	<i>0c (black)</i>
<i>Screen</i>	<i>Tutor1</i>	<i>Any wipe</i>	
<i>Video</i>	<i>Tutor1</i>	<i>FadeOut</i>	
<i>BColor</i>		<i>Any wipe</i>	<i>0c (black)</i>
<i>Screen</i>	<i>Tutor2</i>	<i>Any wipe</i>	

---

## ***FadeOUT***

---

This method allows you to fade an image out to black. (This feature may not work on all VGA cards.)

### ***Type***

Select Video from the Type menu.

### ***Method***

Select FadeOUT from the Video menu. The cursor goes to the File field.

### ***File***

The image that is faded out is designated in the File field and it should be the current image on the graphics screen. This event only affects the palette of the image.

## ***ViewPort***

---

ViewPort allows you to create an “active zone” in your visible graphic screen. This zone becomes the **only active area on the screen**. You can move sprites through it, wipe images into it, and so on. Everything that is displayed on the graphic screen shows up only if it is within the active viewport.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select ViewPort from the Video menu. You must reset the viewport to include other areas of the screen if you want to use other parts of the screen after you have set the viewport. You can turn the viewport off with ViewOFF.

### ***Parameters***

When you select ViewPort, the menu bars disappear. A message displays in the upper left corner telling you to set the lower left corner of the viewport and press Select or [Enter]. A ghost box appears. Using the mouse, size the

viewport the way you want it and press Select. The parameters of the viewport show in the Parameters field.

---

*NOTE: When using a viewport in the 16- and 4-color modes, the screen location of some sprites may change. This is due to a byte boundary constraint that causes the bytes in the pixels to realign themselves at runtime. Adjust your viewport to accommodate this by making sure the viewport x value is equally divisible by 8.*

---

## **ViewOFF**

---

ViewOFF is used to restore the whole graphics screen as the active area. It turns the viewport off.

### **Type**

Select Video from the Type menu.

### **Method**

Select ViewOFF from the Video menu. Now the entire screen is the active viewport.

## **CEGMODE**

---

CEGMODE is used to activate the Analogue Devices CEG-DAC Continuous Edge Graphics chip. The various modes are set in the Parameters field. You must make sure the mode you set is consistent with the CEG image you are showing. This setting is automatically turned off if you make any subsequent events that change the palette. These events include color cycling, loading or resetting palettes, and playing FLI files.

### **Type**

Select Video from the Type menu.

### **Method**

Select CEGMODE from the Video menu.

### ***Parameters***

The Parameters field is used to set which CEG mode you want to activate. This mode setting must work in combination with the Video mode you have set and the method you used to create the images. For more information on CEG modes and images, please refer to the Edsun documentation and data sheets.

### ***ClockOn***

---

ClockOn allows you to display a system clock with date and time. This time and date information is displayed in the default font set with the Global font event. Be sure the computer you are running has its internal clock set correctly because TEMPRA Media Author uses this internal clock as the source for the date and time information displayed.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select ClockOn from the Video menu.

### ***Parameters***

In the Parameters field, you select the background and foreground colors for the clock. Now you are asked to position the clock on the screen.

### ***ClockOff***

---

ClockOff turns the system clock display off.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select ClockOff from the Video menu.

## **Animation**

---

Animation focuses on bringing animation to a sequence from animation (FLC, FLI or FLX) files. Also included in this section is animation of text and colors on the screen.

## **CycleCol**

---

This method allows you to cycle colors of the image on your screen, in a controlled fashion, using a range of colors in the palette. This event can be used to create animation and interesting psychedelic visual effects.

### **Type**

Select Video from the Type menu.

### **Method**

From the Video menu, select CycleCol.

### **Parameters**

The Parameters field is used to determine the beginning and ending colors and how many cycles. If you have a range of colors to cycle that is 25b 34e, it takes 10 color shifts for the colors to return to the starting place. The number of cycles default is "0" and will make a complete cycle. Color cycling is accomplished by rapidly applying all the colors in the palette to the screen. Color cycling is available on display modes with 16 or 256 colors only.

---

***NOTE:** The color palette is rearranged by the LoadPal and CycleCol events. If you load in a new palette when there is a VGA graphic on the screen, the new palette changes the appearance of the VGA graphic before the next screen can be loaded. If you don't do a complete cycle, the colors in later events may not appear as you expect.*

---

### ***AAPlay***

AAPlay is used to play FLI, FLC and FLX files made with animation programs like TEMPRA Turbo Animator, Autodesk Animator, Animator Pro and 3D Studio. Once the animation is made, you can show it with TEMPRA Media Author. You can pause the animation by pressing the space bar and start it again by pressing [Enter].

#### ***Type***

Select Video from the Type menu.

#### ***Method***

Select AAPlay from the Video menu. The field cursor goes to the File field.

#### ***File***

Select your file from the list. The default directory for FLI, FLC and FLX files is \TEMPRA\PIC. Running an animation file changes the current palette to the one used in the animation file, even after the line has finished playing. This happens when playing animation files in any 256-color mode.

To reset the video mode, use another Video event with a SetMode method. Most animation files are created in 320x200 resolution. However, animation files can be played in 640x480, 600x800 and 1024x768 resolutions as well. If you need to change the video mode to accommodate playing animation files, select SetMode from the Video menu, and make your choice from the SetMode menu.

#### ***Parameters***

The Parameters field holds the location on the screen where you want to play the animation file. A ghost box appears; place the box on the screen where you want the animation to play. Press Select to secure the box.

#### ***Sp/Dp***

The Sp/Dp field is used to govern the speed at which the animation file plays. The slowest speed is 1, the fastest is 99. Animation files can be made Permanent, Temporary, or Erasing. This setting effects your current palette

because FLI, FLC and FLX files have their own palettes. If you make this a Permanent or Temporary event, the palette changes. If you choose Erasing, the old palette is restored before the next screen is displayed.

## ***Scrollup***

---

Scrollup is used to scroll a text file up the video screen. Text is scrolled in the Global font. You can change the font by selecting a Global event and choosing Font from the Global menu. The smaller the font, the smoother it scrolls. Use of large fonts is discouraged as the scrolling text jumps.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select Scrollup from the Video menu and the cursor goes to the File field.

### ***File***

The file directory of TXT files pops up. Select the text file you want to use. The default directory for text files is \TEMPRA\TXT. Be sure your text files are in this directory or change the default directory by selecting the Path button on the Menu Bar.

### ***Parameters***

The Parameters field is used to set up the scrolling text. When you enter the Parameters field as you make the event, you go through all of the choices automatically. When you edit the Parameters field, you have the choice of which part you want to edit.

### ***All***

This selection allows you to edit all of the Parameter fields.

### ***Position***

Use the mouse or cursor keys to place the text on the visible screen. Be sure to keep all of the text on the screen in the active viewport. If you have not set a viewport, the entire screen is the active viewport.

### ***Color***

Select the color for the text. If you select “Yes” at the “Choose Color from Image?” menu, move the mouse until the cross-shaped cursor is over the color you want, and press Select or [Enter]. If you select “No,” the color palette appears. Select the color you want from the palette and select OK. The background color is set with a Global event, select Color from the Global menu, and select the text background color you want from the palette. Color 0 is good for video overlay (transparent).

### ***Width/Height***

Set the width and height with the mouse or cursor keys by stretching the box around the text on the screen. While you size the box with the mouse or cursor keys, notice that it moves in character-based increments. This allows sizing by number of characters wide and number of lines high.

### ***Frame***

This selection allows you to place the scrolling text inside a bordered box or inside a box without a border. If you select a bordered box, the scrolling text appears in a rectangle of the color set in the Global event. To change the colors of the framed box, select Global from the Type menu, select Color from the Global menu. A box appears in the upper left corner of the screen. Make your color selections and then select Done. If you select no border, the scrolling text appears in a rectangle of the text background color set in the Global event described above.

### ***Justify***

You can choose to have the lines of text justified to the Right, Left, or Center. Please keep in mind that TEMPRA Media Author thinks spaces are characters, so if the justification does not look as you expected, check the end of the lines of text to see that there are no spaces.

### ***Sp/Dp***

The Sp/Dp field determines how fast the text scrolls up the screen. A slower speed is recommended so the text scrolls smoothly and at a readable speed. Scrolling text can be made Permanent, Temporary, or Erasing.

---

**NOTE:** *When you use scrolling text over a graphic image or animated file, the text scrolls over a rectangle of color, with or without a border.*

---



Sharp Boy says:  
Create neat credit effects  
over live or taped video. We  
tell you how.

If you want to use scrolling text over live or taped video (as in credits), you can have it appear without the background color (transparent) by following these steps before you bring in the text file:

1. Select Global from the Type menu; select Color from the Global menu; and set the text background color to 0 (transparent black).
2. Select Video from the Type menu; select VidTcol from the Video menu; and set the transparent color to 0, which is the default setting. If your live or taped video is running in the background when you write this event, you may see the video running through blocks in the color palette. If so, you can select one of those squares as your transparent color.

Now the scrolling text will appear directly on top of the live or taped video.

## ***Ticker***

---

This method is used to scroll text horizontally across the screen, giving a “ticker tape” effect.

### ***Type***

Select Video from the Type menu.

### ***Method***

When you select Ticker from the Video menu, the cursor jumps to the File field.

### ***File***

A list of TXT files appears on the screen. Choose the file you want to use. The text can be any place on the screen, any length that will fit on the screen, and the text can be any color in the palette. The font is set by selecting Global from the Type menu, selecting Font from the Global menu, choosing which font you want to be the default.

### ***Parameters***

The Parameters field is used to set up the Ticker display. When you enter the Parameters field when you first make the event, you go through all of the choices automatically. When you edit the Parameters field, you can select which field you want to edit.

### ***All***

This selection allows you to edit all of the Parameters fields.

### ***Position***

Use the mouse or cursor keys to locate the Ticker display on the visible screen. Be sure to keep all of the text on the screen in the active viewport. If you have not set a viewport, the entire screen is the active viewport.

### ***Color***

Select the color for the text. If you select “Yes” at the “Choose Color from Image?” menu, move the mouse until the cross-shaped cursor is over the color you want, and press Select or [Enter]. If you select “No,” the color palette appears. Select the color you want from the palette and select OK. The background color is set with a Global event. Select Global from the Type menu, select Color from the Global menu, select the text background color you want from the palette.

### ***Width***

Set the width with the mouse or cursor keys by stretching the box around the text on the screen. While you size the box, notice that it moves in character-based increments. This allows sizing by number of characters wide. If you change fonts, you may need to change the width of the box.

### ***Frame***

You can choose to have the ticker display appear inside a bordered box or with no border. The text appears in a rectangle of color set in a Global event. Select Global from the Type menu; select Color from the Global menu; select the background and border colors you want from the palette.

**Justify**

Does not apply to a Ticker event.

**Sp/Dp**

The Sp/Dp field sets the speed the text moves across the screen. A slower speed is recommended so the text scrolls smoothly and at a speed that is easy to read. Scrolling text can be Permanent, Temporary, or Erasing.

**Motion Video**

---

The following Video Window events require a video card (such as a SNAPplus or Video Blaster) to be installed. To get video to the screen, use the VidWin method.

**VidLive**

---

VidLive turns on live or taped video and displays it in the last video window that was used. If there is no video window set, this function will not work. VidLive does not have parameters.

**Type**

Select Video from the Type menu.

**Method**

Select VidLive from the Video menu.

**VidOver**

---

The VidOver method places the video in overlay mode and fits into the last video window. The video coming into the screen is only seen when there is a transparent color on the display. VidOver is different from the VidLive method, which places the video on the screen on top of any graphics.

**Type**

Select Video from the Type menu.

### ***Method***

Select VidOver from the Video menu. To set the transparent color to a color other than 0, select Video from the Type menu; select VidTcol from the Video menu; and select the transparent color. Wherever the selected transparent color appears on the screen, video will be displayed in its place.

### ***VidNorm***

---

The VidNorm method turns off any video window that is active. The display returns to a normal VGA monitor. You should use VidNorm whenever you do graphic events in a sequence following a live or taped video event.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select VidNorm from the Video menu. The live or taped video parameters previously set are now turned off.

### ***VidWind***

---

The VidWind method tells TEMPRA Media Author to place a video window onto the screen and display the video coming into the video card. The video is brought into TEMPRA Media Author by a compatible display card. If your computer does not have a compatible display card, VidWind will not function and you will receive an error message.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select VidWind from the Video menu.

### ***Parameters***

The Parameters field allows for placement of the video window. You are asked to position the video window as you create the event. The screen clears and asks you to place the lower left corner of the video window. Use

the mouse to position the lower left corner of the window and press Select; then move the mouse to outline the area where you want the video window to appear. As you move the mouse a rectangle appears; use this rectangle to place and size the window; then press Select on the mouse or [Enter] on the keyboard. The coordinates of the video window appear in the Parameters field.

## ***VidGrab***

---

The VidGrab event freezes the video coming into a video window and places it onto the screen. The image then becomes a part of the screen display. Captures can appear to be instantaneous. Depending on the type of video input board, the grab time can vary from 1/30th of a second to a few seconds. You have to be in 16-bit, high-color mode for VidGrab to work.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select VidGrab from the Video menu.

## ***VidTcol***

---

VidTcol allows you to select any single color to be a transparent for showing video in a video window. The video coming into the computer is only visible where the transparent color is on the screen.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select VidTcol from the Video menu. When a video window has been established on the screen and a video event using VidTcol is performed, any further events that bring the transparent color to the screen will bring the video to the screen through the transparent color. These events can be Screen, Sprite, or BColor (background color) events.

### ***Parameters***

The Parameters field allows you to choose the transparent color. Your choice may be from the screen or from the color palette. If you are using a 256-color video display, you should load the palette of one of your images before you pick the transparent color. Select Video from the Type menu; then select LoadPal from the Video menu. A list of images will pop up. Choose one of the images you will use in your sequence.

### ***VidBW***

---

VidBW causes the video coming into the computer to switch between black and white and color.

#### ***Type***

Select Video from the Type menu.

#### ***Method***

Select VidBW from the Video menu. You are asked if you want to turn VidBW on or off. Once you have executed an event that changes video to black and white, you must execute another event that turns VidBW off to return the video display to color.

### ***VidInv***

---

VidInv inverts the colors of the video coming into the display through a video window.

#### ***Type***

Select Video from the Type menu.

#### ***Method***

Select VidInv from the Video menu. You are asked if you want to turn VidInv on or off. Once you have executed an event that changes video to Inverse, you must execute another event that turns VidInv off to return the video display to normal.

## ***Infowindow***

Infowindow applies only to IBM Infowindow touch screen capabilities.

## ***Infowind***

Infowind is used with a variety of parameters to control the output of an IBM Infowindow. You can use it to show graphics according to the mode you are in. (There is no VGA in the Infowindow.) You can use it to show video coming through the Infowindow.

### ***Type***

Select Video from the Type menu.

### ***Method***

Select Infowind from the Video menu.

### ***Parameters***

When you select Infowind, the field cursor goes to the Parameters field.

### ***Video***

Select this parameter to make the IBM Infowindow display show Video from an external source. If there is no video coming into the Infowindow display, executing this event blanks the screen.

### ***Graphics***

Select this parameter to make the IBM Infowindow display show graphics from the PC. You can show CGA and EGA graphics with an Infowindow, but you cannot show VGA. This is a limitation of the Infowindow graphics hardware. Be sure you have selected the correct video mode.

### ***Overlay***

Select this parameter to mix computer output with video coming in from an external source on the IBM Infowindow display. You can show 80 column text, CGA, and EGA graphics with an Infowindow, but you cannot show VGA. This is a limitation of the Infowindow graphics hardware. Be sure you have selected

the correct video mode. Black (color 0) is the default invisible color so every pixel that is black becomes transparent.

### ***Reset***

Selecting this parameter resets the Infowindow to a default condition. The default condition results in: auxiliary output enabled, transparency disabled, normal palette, normal graphics and video, composite input 1 enabled, and interlace disabled. Please refer to your Infowindow documentation for more information on reset.

## Chapter Example

This chapter example sequence is available in the \TEMPRA\SEQ directory and is named VIDEO.SEQ. Below is a copy of the Workform and a brief explanation of each line.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Video		SetMode		SGA_2E 640x480 256		
	Video	TUTOR1	LoadPal				
	Global	COUR20	Font				
	Global		PIC_PCX				
	Global		Colors		47 167B 191 0T 15 4H		
	Video	SCROLL	Scrollup	2 E	179x 385y 48c 21w 8h nrm	5 0/18	
	Text	VIDEO	FUTURA20	E 013	87x 355y 10c 21w 9h nrm	8 0/18	
	Text	VIDEO	TIMES25	E 022	37x 298y 127c 24w 7h nrm	5 0/18	
	Screen	TUTOR1	Drip_DN	99			
	Video	TUTOR1	FadeOUT	99			
	Video	TUTOR1	FadeIN	99			
	BColor		Dissolve	99	Solid 0c		
	Text	VIDEO	TIMES25	E 028	10x 360y 153c 25w 13h nrm	8 0/18	
	BColor		Lft_Rt	99	Solid 0c		
	Video	TOUCAN2	AaPlay	99 P	0bx 407by 2R		
	Video	TOUCAN2	AaPlay	99 P	293bx 329by 2R		
	BColor		Dissolve	99	Solid 6c		
	Text	VIDEO	TIMES20	E 040	86x 333y 234c 18w 10h nrm	5 0/18	
	Screen	SLIDE20	Box_In	99			
	Video		CycleCol	99	59b 150e 0c		
	BColor		Box_In	99	Solid 0c		
	Video	TICKER	Ticker	2 P	83x 285y 48c 26w 3h n		

The first two lines are Video events. The first sets the video mode and the second loads the palette from TUTOR1.PCX so the palette won't shift when we bring in our image.

The next three lines are Global events to set the font to COUR20; to select PCX files for our Screen events; and to choose our colors.

Now we have a Video event called Scrollup. This method brings an ASCII text file called "SCROLL.TXT" to the screen, moving up from the bottom of the screen. Notice the Sp/Dp and Wait fields. For a smooth scrollup, a slower speed is best. This also makes the text easy to read. Scrollup uses the default font set in the Global event on the third line. The Wait is used to give the viewer time to finish reading the text before it leaves the screen.

The next two lines are Text events using a file called "VIDEO.TXT" that we created in our text editor. Again, notice the Wait field and our use of the Wait for Timeout choice, which keeps the text on the screen for a set amount of time.

Now a Screen event brings our image (TUTOR1.PCX) to the screen using the Drip\_DN method. Drip\_DN takes a few moments so we've set the speed at 99 which is the fastest possible.

Once the image is on the screen, our next Video event begins to fade it out to black using the FadeOUT method. Then FadeIN takes over and brings the image back to the screen. (See the documentation on FadeOUT and FadeIN before you create your own sequences using these methods.)

Next is a BColor event used to clear the screen.

The next Text event line uses the same text file called "VIDEO.TXT." As you create your sequence, you can select which lines of the text file you want to use.

Another BColor event clears the screen. Notice we have used color 0c (transparent black) as we want the screen to go to black.

Next we have two Video event lines using the AaPlay method to bring animation to the screen using the TOUCAN2.FLI file. You can set the number of repeats in the Parameters field. We've set the repeats at 2 (shown in the Parameters field as 2R).

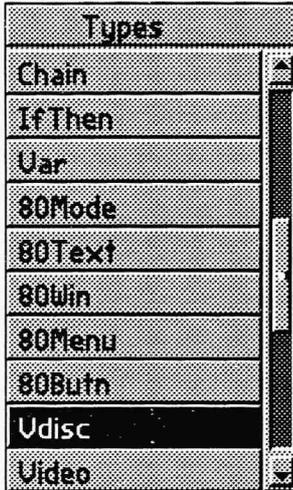
We clear the screen again with a BColor event.

Now we have used another Text event using the same VIDEO.TXT file. Notice that you can use any font, color and position for your text event even though you use the same text file.

The Screen event brings the SLIDE20.PCX file to the screen using the Box\_In wipe.

The next event is a CycleCol method from the Video menu. The first two numbers in the Parameters field are color numbers. The third number (0c) is the number of cycles. Zero is the default and makes a complete cycle between the two color numbers chosen.

The last Video event is the Ticker method. We created the text file named "TICKER.TXT" with our text editor. You will notice the speed in Sp/Dp is set at 2. The slow speed is necessary for the viewer to read the file. Ticker uses the default font set in the Global event on Line 3.



## Chapter 36

# A Vdisc Event

Instant access to high-quality motion video comes only from videodiscs. The Videodisc event is used to operate videodisc players from within TEMPRA Media Author. The players that can be operated are the Sony 2000 and the Pioneer 6000 and 8000. With this feature, you can search for frames, play motion video, play audio independent of the video, and more, depending on the capabilities of the videodisc player.

---

*NOTE: If your videodisc player is command compatible with one of the players mentioned above, but TEMPRA Media Author doesn't work, check the BAUD rate for your player and set the Environmental Variable VDISC\_BAUD to your BAUD rate. Refer to the CONFIGMA to select the player and COM port for your player.*

---

## Type

This event requires you to have a videodisc player connected to the computer on which TEMPRA Media Author is running. The Pioneer 6000 and 8000 or Sony 2000 can be connected via the RS232C serial connection. The Videodisc event itself controls access to the sound channels, frames, and video scenes on the videodisc.

## **Method**

You can perform the following functions on the videodisc through the computer.

### **Playseg**

Play a video segment from beginning to end.

### **PlaysegP**

Play a segment allowing for the user to stop and restart the segment with space bar and [Enter], and to leave the segment with the [Esc] key.

### **Search**

Go directly to a frame.

### **Stop**

Stop the forward or reverse play of the video segment.

### **PlayFWD**

Play forward from the current frame, until further notice.

### **PlayBKW**

Play backward from the current frame, until further notice.

### **FrameDis**

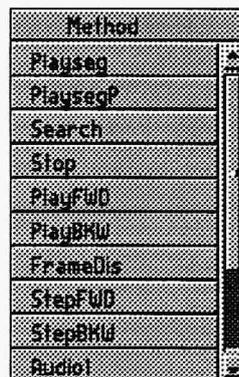
Display the frame number on the videodisc.

### **GetFrame**

Returns the current frame number to a numeric variable, which can then be analyzed with the IfThen statement.

### **StepFWD**

Advance one frame on the videodisc.



**StepBKW**

Retreat one frame on the videodisc.

**Audio1**

Control the audio from videodisc audio track 1.

**Audio2**

Control the audio from videodisc audio track 2.

**Sendcom**

Sends the contents of the Parameters field out the serial port as a string.

**Parameters**

---

The entries in the Parameters field are tied to the method choice. The ones that need parameters are:

**Playseg**

Start and end frame numbers.

**PlaysegP**

Start and end frame numbers.

**Search**

Frame number.

**FrameDis**

On or Off (for Frame Number Display). This is most useful while you are authoring and need to see the frame number for each frame.

**GetFrame**

The Var name to store the frame number.

### ***Audio1 and Audio2***

On or Off (to control the audio tracks).

### ***Sendcom***

The ASCII string you want to send to the player.

### ***Chapter Example***

---

This example tells the videodisc player to search to frame number 12345.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Vdisc	Search			12345		

---

# ***Appendices***

- A***    *Keyboard Functions*
  - B***    *TEXTEDIT Information*
  - C***    *Using Serial I/O*
  - D***    *Creating Music*
  - E***    *PWin Utility*
  - F***    *Keyboard Return Codes*
  - G***    *Font Selections*
  - H***    *Technical Information*
  - I***    *TEMPRA SHOW Tutorial*
  - J***    *TEMPRA Media Author Tutorial*
-

# Appendix A

## Keyboard Functions

<b>INTERFACE</b>	<b>KEY</b>	<b>MEANING</b>
Graphic/Workform	Backspace	Erases contents of field under cursor
Graphic/Workform	Home	Go to beginning of line
Graphic/Workform	End	Go to end of line
Graphic/Workform	Left Arrow	Move left
Graphic/Workform	Right Arrow	Move right
Graphic/Workform	Up Arrow	Move up
Graphic/Workform	Down Arrow	Move down
Workform	PgUp	Go up a page
Workform	PgDn	Go down a page
Graphic/Workform	Ctrl-PgUp	Go to top of form
Graphic/Workform	Ctrl-PgDn	Go to bottom of form
Graphic/Workform	Ctrl-Rt Arrow	Word/field right (character edit)
Graphic/Workform	Ctrl-Lft Arrow	Word/field left (character edit)
Graphic/Workform	Esc	Escape current choice
Graphic/Workform	Enter	Accept current choice
Graphic/Workform	Ins	Insert line
Graphic/Workform	Ctrl-BkSp	Erase line
Graphic/Workform	Del	Delete line
Workform	F1	Help index
Workform	F2	Text editor
Graphic/Workform	F3	Edit field
Graphic/Workform	F4	Search/Replace string
Graphic/Workform	F5	Preview line
Graphic/Workform	F6	Preview from line
Workform	F7	Mark block begin
Workform	F8	Mark block end
Workform	Alt-C	Copy block
Workform	Alt-D	Delete block

## Keyboard Functions

---

<b>INTERFACE</b>	<b>KEY</b>	<b>MEANING</b>
Graphic/Workform	Alt-G	Go to line
Workform	Alt-H	Hide help info
Graphic/Workform	Alt-I	Information on system state
Graphic/Workform	Alt-K	Delete text to right of cursor
Workform	Alt-L	Load block
Workform	Alt-M	Move block
Workform	Alt-S	Save block (cursor shows)
Graphic/Workform	Alt-S	On menus, sorts the menu
Graphic/Workform	Alt-T	Toggle touch screen and mouse
Workform	Alt-U	Unmark block
Graphic/Workform	Alt-X	Toggle output screen
Graphic/Workform	Alt-Z	Turns on trace
Graphic/Workform	Ctrl-A	Audio menu
Graphic/Workform	Ctrl-D	Disk/Print menu
Graphic/Workform	Ctrl-G	Toggle between GUI/Text interface
Graphic/Workform	Ctrl-L	Load a sequence
Graphic/Workform	Ctrl-Z	DOS Shell (type EXIT to return)
Graphic/Workform	Ctrl-N	Start new sequence
Graphic/Workform	Ctrl-P	Path menu
Graphic/Workform	Ctrl-Q	Quit
Graphic/Workform	Ctrl-S	Save a sequence
Graphic/Workform	Ctrl-X	Toggle x,y coordinate display

## ***Runtime Hot Keys***

### ***Forward and Back***

While you are running a sequence, you can back up label-by-label by pressing the [F9] key. Every time you press the [F9] key you can back up one label. Then you can restore your position label-by-label by pressing [F10]. If you press any other key, you lose the ability to skip ahead label-by-label.

To use this effectively, you should label the important screens in your presentation by entering a name in the Label field for each of these screens. You can back up to the most recent 50 labels. Pressing [F10] allows you to restore your previous position one label at a time. When you press [F10] and you have not yet backed up, you skip through one event line at a time.

This feature is automatically on when you start the program. You can turn it off by making a Global event, BackOff method.

### ***Emergency Key - Home Base***

Sometimes it is a good thing to be able to get to a predefined place, a kind of Home Base, in your presentation from any place you happen to be. You can do this by making a special label. This label, <F1>, written in the Label field allows you to jump to this line from anywhere just by pressing the [F1] key.

### ***Use the Global Subroutine Key***

Mark a line with <F2> in the Label field. At the end of the subroutine, put <Ret> in the Next field. When the user/viewer presses [F2], the system will look for that label in the sequence.

## **Appendix B**

# **TEXTEDIT Information**

The Word button on the Menu Bar starts your DOS text editor. TEMPRA Media Author is shipped with a default text editor called TEXTEDIT. If you selected this text editor program during installation of TEMPRA Media Author, the following command descriptions will help you use TEXTEDIT.

1. Select the Word button. A menu of files in the default directory will appear on the screen. The default directory is the TEMPRA directory.
2. If your text files are stored in a different directory, press [Esc] to bypass the menu. A line will appear at the bottom of the screen "File does not exist, Do you want to create a new file? (Y/N)."
3. If you select "Yes," you will see a blank screen ready for your text input.
4. If you select "No," you will be prompted to type in the name of the file. You must include the path.

### **TEXTEDIT Commands**

#### **Control S = Save**

This function allows you to save your text file. The default text directory in TEMPRA Media Author is \TEMPRA\TXT. To save your text files to the TEMPRA\TXT directory, you must type in the path as well as the document name. You now have a clear screen and are ready to create another file.

**Control Q = Quit**

A box will appear titled "Exit to DOS?". "Yes" will return you to TEMPRA Media Author. "No" leaves you in TEXTEDIT.

**Control L = Load**

This function allows you to load an existing file by typing in the path and filename.

**Command Description**

**Key Stroke**

Backspace or delete previous character	Backspace
Carriage return or next line	[Enter]
Change a string (All,Range,Block)	F6
Change a string with verify (All,Range,Block)	Shift-F6
Copy a highlighted block of text	Alt-C
Delete a highlighted block of text	Alt-D
Delete current character	Del
Delete current line	Shift-F9
Delete from current position to end of line	Alt-K
Find a string (All,Forward,Backward)	F5
Goto a line number	Alt-G
Help Screens	F1
Highlight a line block	F8
Highlight a character block	F7
Insert a line after the current line	F9
Insert a line before the current line	Ctrl-F5
Join next line at the end of the current line	Ctrl-J
Load file	Ctrl-L
Memory status display	Alt-I
Move a highlighted block of text	Alt-M
Page down	PgDn
Page up	PgUp
Position at the beginning of the file	Ctrl-PgUp
Position at the end of the file	Ctrl-PgDn
Position at the beginning of the line	Home
Position at the end of the line	End

***Command Description******Key Stroke***

Position down one line	Down arrow
Position up one line	Up arrow
Position on the next character	Right arrow
Position on previous character	Left arrow
Position on the next word	Ctrl->
Position on the previous word	Ctrl <-
Quit the current file	Ctrl-Q
Save the current file and continue editing	Ctrl-S
Tab toward right	Tab
Tab toward left	Shift-Tab
Toggle between insert/replace mode	Ins

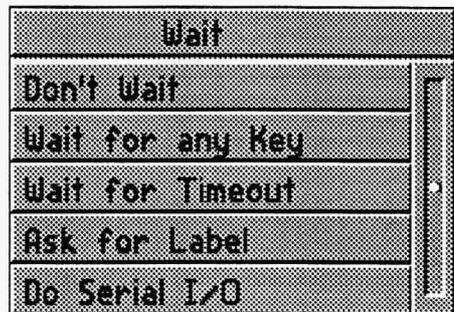
## Appendix C

# Using Serial I/O

Serial I/O or, input and output through the serial ports, can be accomplished by selecting “**DO SERIAL I/O**” from the “**Wait**” menu. This feature can also be used to send data through the parallel ports.

This feature has been put in the Wait field because we determined that, if you are going to be using Serial communications, you would want to receive or send a value through the serial port before you went to the next line.

By using this feature, you can create and transmit (or receive) values in a single line.



The values sent through the serial port are string or numeric variables. First the Var must be declared, then modified according to the command syntax that follows, or the Vars can be read from text files.

To create a value, you can build the string by adding small string values to the declared variable. In this case, you should use the += modification operator followed by the string constants you want to add.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Modify		\$com += "I"		
	Var		Modify		\$com += ",1"		
	Var		Modify		\$com += ",1200;"	\$com	

The sequence above will initialize serial port 1.

Otherwise, you can make the string Var equal to a constant string by setting the String Variable equal to a constant string.

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Modify		\$com = "I,1,1200;"	\$com	

The sequence above initializes port 1. (See Syntax below)

After the communication has occurred, it is possible to test the value of the received or resulting information by using the IfThen event and testing for the value of the numeric Variable #\_com1,#\_com2....#\_comB.

Serial I/O does not send carriage return or line feed information. If you need to end a transmission with <<CR>>, send a "13" for the carriage return, and "10" for line feed <<LF>>.

## ***Command Summary***

---

The commands must be built according to the syntax that follows. Commands can be sent individually, or more than one command may be sent at a time, but the commands must be separated by semicolons.

### ***Initialize***

**I,port,parameter;port number =1-11,A-D**

<u>Port Number</u>	<u>Base Address</u>	<u>Result</u>
<b>Serial Ports</b>		
port 1	0x3f8	#_com1
port 2	0x2f8	#_com2
port 3	0x2e8	#_com3
port 4	0x2e0	#_com4
port 5	0x3e8	#_com5
port 6	0x3220	#_com6
port 7	0x3228	#_com7
port 8	0x4220	#_com8
port 9	0x4228	#_com9
port 10	0x5220	#_comA
port 11	0x5228	#_comB
<b>Parallel Ports</b>		
port A	lpt1	#_lpt1
port B	lpt2	#_lpt2
port C	lpt3	#_lpt3
port D	lpt4	#_lpt4

When parameter is a number or numeric variable, TEMPRA Media Author reads it as the baudrate, or bits per second (bps = 110, 150, 300, 600, 1200, 2400, 4800, 9600), and always uses 8 data bits, 1 stop bit, and no parity.

When parameter is a string variable of the form:

**“bps,parity,databits,stopbits”**

Bps is the baud rate.

Parity is O=odd

E=even

N=no parity

Databits is either 7 or 8

Stopbits is either 1 or 2

Parameter is ignored for parallel operations.

## **Send**

**S,port,value;**  
port = 1-11 or A-D  
value = 0 - 255 (decimal)  
value can be a numeric var #  
value can be a string var \$ (for serial only)

## **Receive**

**R,port,flag;**  
port =1-11;  
the received value is in #\_comX  
flag =1 — wait until received  
flag =0 — return after trying to receive  
#\_com1 #\_com2 ... #\_comB

TEMPRA Media Author creates this variable and the value is modified after the data is received.

The values received with the “R” command are numeric and are declared by the TEMPRA Media Author system. The received variables are #\_com1, #\_com2, #\_com3... #\_comB. One value is for each serial port. The value is changed every time a communication is successful.

### **Internal Variables for Receive**

#\_comX = value

Here X is equal to the serial port from which the data came. Value is the returned byte (0-255), or -1 (if flag on Receive was 0 and nothing was there). If you want to do this, your cable will have to have the pin connections listed below.

## **Get**

```
G,port,terminator;  
port =1-11
```

The terminator value is a decimal ASCII value.

The **Get** command stops when the terminator value is received, or the length is 255 characters long, or a timeout occurs. Time out occurs after 24,000 reads. The string goes into Vars: \$\_com1, \$\_com2, ... \$\_comB.

TEMPRA Media Author creates these variables and the value is modified after the data is received. The values received with the “G” command are string and are declared by the TEMPRA Media Author system. The received variables are \$\_com1, \$\_com2, \$\_com3... \$\_comB. The strings can be up to 255 characters long. The value is changed every time a communication is successful.

### **Internal Variables for Get**

```
$_comX = value
```

Here X is equal to the serial port from which the data came. Value is the received string up to 255 characters long.

### **Serial Cable:**

PIN	PIN
1 -----	1
2 -----	3
3 -----	2
7 -----	7

### Parallel Cable:

PIN .....	Data Bit
2 .....	0 (possible values: 1 = +5ttl, 0 = neutral)
3 .....	1
4 .....	2
5 .....	3
6 .....	4
7 .....	5
8 .....	6
9 .....	7
18-25 .....	ground

### ***Notes on Parallel Transmissions***

The parallel port operations allow you to send data through the port. It actually sets lines “high” according to the value you send. This type of communications is not “intelligent” enough to do much more than that. At the conclusion of a parallel communication, you get the status of the port in the following internal Vars: #\_lpt1..#\_lpt4. These values are based on the standard status flags of the parallel device you are sending to. For example, “printer not ready” may be the meaning of one of the values you receive.

### ***Reading Commands from a Text File***

Instead of building commands in TEMPRA Media Author the designer can make a text file with all the commands and read them in as needed.

Serial I/O command files have the following syntax:

```
command,param1,param2;command,param1,param2;
```

**EACH COMMAND MUST END WITH A SEMICOLON!!**

## **Appendix D**

# **Creating Music**

Music text files can be created and edited prior to, or while, running TEMPRA Media Author with any text editor or word processor that can operate in non-document (ASCII) mode. Be sure to save your Music text file (suffix.MUS) to the proper directory. The default directory in TEMPRA Media Author is C:\TEMPRA\AUD.

### **MUSIC SYNTAX**

Music is composed by creating a text file with music commands or by using the appropriate method and putting the information in the Parameters field.

### **MUSIC COMMANDS for PLAY and PLAYFILE**

The format for Play and PlayFile is the same; the only difference is that the PlayFile method takes its information from a text file. The Play commands use what you have entered in the Parameters field.

### **FOR OCTAVES**

- OX** (x is in range 0-9) Set the OCTAVE for subsequent notes.  
Example: O9 C  
Result: Plays a high "c"
  
- >** Increase the current octave  
Example: O2 C > C  
Result:: Plays a "c" and then a "c" one octave higher.
  
- <** Decrease the current octave  
Example: O2 C < C  
Result: Plays a "c" and then a "c" one octave lower.

## **FOR NOTES**

**Ax-Gx** (Notes A through G; x is in range 1-64) Play the indicated note in the current octave and time. Change L to change the default note length. Change x to change this note's length.

x = 1 Play a whole note

x = 2 Play a half note

x = 4 Play a quarter note

x = 8 Play an eighth note

Example: O2 C1 < C2

Result: Plays a whole note "c", then a half note "c" one octave lower.

**#,+** Make the preceding note a sharp

Example: O2 C# < C

Result: Plays a "c" sharp, then a natural "c" one octave lower.

**-** Make the preceding note a flat

Example: O2 B- < B

Result: Plays a "b" flat, then a natural "b" one octave lower.

## **FOR LENGTH**

**Lx** Set the default note length (x is in range 1-64)

Example: O2 L4 B < B

Result: Plays a quarter note "b" and then a quarter note "b" one octave lower.

A period after a note extends that note's duration by 50%.

Example: O2 L2 B. < B

Result: Plays a note "b" for 3 counts and then a note "b" for two counts (half note) one octave lower.

**MS** Staccato - play notes that follow for 3/4 of their current length.

**MN** Normal - play notes that follow for 7/8 of their current length.

**ML** Legato - play notes that follow for their current length.

## **FOR TEMPO**

- Tx** (x is in range 30-255) Sets the Tempo based on the number of 1/4 notes in a minute. The default is 120.
- Px** (x is in range 0-255) Sets pause in the music with x being the length of the pause.

## **FOR BACKGROUND/FOREGROUND PLAY**

Enter these characters in your text file.

- MF** Play the following notes in the foreground.
- MB** Play the following notes in the background.

---

***NOTE:** The Music background command should only be used to play a single line. If you try to play a number of lines with a large Loop, it may be a while before control returns. One line is approximately 215 characters in length.*

---

For a more complete explanation consult a music textbook.

## **MUSIC COMMANDS FOR TONE AND TONEFILE:**

- fd** Play a tone of frequency f for d milliseconds.

## **MUSIC COMMANDS FOR FREQ AND FREQFILE:**

- fd** Set the speaker frequency to f for d milliseconds.

---

***NOTE:** The format for Tone/Freq and ToneFile/FreqFile is the same, the only difference is the ToneFile/FreqFile methods take information from a text file. The main difference between a Freq and a Tone is that when you play a Tone, you automatically set the frequency, turn on the speaker, wait, and then turn off the speaker. Using Freq allows you to vary the frequency without turning the speaker on and off.*

---

## Creating Music Example



This is an example of a Concerto for Trumpet. If you type it out, save it as a MUS file, then copy it to the \TEMPRA\AUD subdirectory, you can use it in a Music event.

```
t50o2f4.a16g16g8c8p8>c8c8<b-16a16b-8b-8  
b-8a16b-16>c16<b-16a16g16f4p8a16g16g8c8  
>c4.d16e16f8e16d16e16d16c16<b16>c4.d16  
e16f8e16d16e16d16c16d16<b64>c64<b64  
>c64<b64>c64<b64>c64<b64>c64  
<b64>c64c16c4
```

A brief explanation of some of the above text file entries follows.

t is the Tempo (50 beats per minute).

o2 is the octave 2.

Period in f4. extends the note by 50% (dotted quarter).

a16 is a 16th note (a)

>c8, the > takes you up an octave and

< brings you down an octave.

- in b- is a flat; a # would be a sharp.

The 12 64th notes make a dotted eighth trill.

## ***Appendix E***

# ***PWin Utility***

PWin is a program developed specifically for the creation of text windows that have text graphic characters and that use color. These are the 80Win \*.WND files. The intended use for this kind of text window is the creation of screen forms. They can also be used in the development of screens that have to be shown in text mode, but which also require graphics characters. You can use the graphics text characters in the ROM.

### ***Getting Around in PWin***

Use a mouse or the cursor keys to move the cursor. For Menu Selections: move the highlight bar on the main menu over the choice you wish to make and press [Enter]. Use a mouse or the cursor keys to move the cursor over the screen.

For large cursor movements:

[Home]	to move the cursor to the left column position.
[End]	to move the cursor to the far right cursor position.
[PgUp]	to move the cursor to the top of the screen.
[PgDn]	to move the cursor to the bottom of the screen.

### ***Starting PWin***

To start PWin, at the DOS prompt type "PWIN," then press [Enter]. When the program starts, press [Enter] to clear the copyright screen. Now you can see the Options Menu.

Edit Window	Edit contents of a WND file.
Set Color	Set the colors for foreground and background.
Choose Character	Choose the graphic character to put on the screen.
Box	Create a bordered or non-bordered box.
Read WND	Read in a WND file that has been made with PWin.
Read SCR	Read in a SCR file that was grabbed from the screen.
Write WND	Save the contents of the screen..
Toggle Status Line	Choose to display status line while editing.
Help	Displays the Help file.
Quit	Leave PWIN.

## ***Edit Window***

This selection is used when you want to enter text into a window from the keyboard. The text you enter will have the color attributes set in the choose color selection from the Main Menu. Every time you press [Enter] the graphic character shown on the status line will be put on the screen. Remember to use the cursor keys or mouse exclusively when moving the cursor on the screen. You do

not have any of the expected text editing functions. What you type is what you get. Be sure to press [F1] to learn what the Help system says. **Be aware that [Ctrl] backspace will clear the screen.**

## ***Choose Color***

The first thing you will want to do is select the current color attributes for the text and graphics characters. All text has two types of color: the foreground color, which we generally think of as the character itself and the background color, the field that the character sits in. The “Sample Text” box shows you what the current text color selections are. Use the left menu to choose which attribute you want to change first, then select the color from the right menu. You will notice that you have more choices of text colors for the foreground than the background. If you want to return to the previous color choice before you Quit, select Cancel. Otherwise, when you Quit, the colors you set will be active for all the text and graphic characters you put in the window.

Set Foreground	Blink On/Off	[ ]
Set Background	Black	[ ]
Cancel	Blue	[ ]
Quit	Green	[ ]
	Cyan	[ ]
	Red	[ ]
	Magenta	[ ]
	Brown	[ ]
	White	[ ]
	Dark Grey	[ ]
	Light Blue	[ ]
	Light Green	[ ]
	Light Cyan	[ ]
Sample Text	Light Red	[ ]
Border Color	Light Magenta	[ ]
Text Color	Yellow	[ ]
Highlight Color	Bright White	[ ]
Text Color		

## **Choose Character**

When you select Choose Character, you will see all the characters in the ROM. These are your graphics characters. Every time you position the cursor with the mouse and press [Enter], the graphics character you select will be placed on the screen.

## **Box**

The second thing you will want to do is create a box for your text. When you select this option, you are prompted to select the upper left corner and the lower right corner of the box. Position the cursor according to the screen prompts: upper left corner and press [Enter], then lower right corner and press [Enter]. Next you will be asked what style of border you want. Make the selection from the menu and you will see the border appear on the screen.

What Type of Border?
Single Border
Double Border
Single Filled
Double Filled
Filled with current character

## **Read WND**

Select this to load a WND file made with PWIN. Before you load a window you want to make sure it is going to load where you want it. Designate this spot by moving the cursor to the spot and pressing [F2]. This will anchor the upper left corner of the window when it loads.

## **Read SCR**

Select this to load a SCR file grabbed with the GRAB utility. Before you load this file, you want to make sure it is going to load where you want it. Designate this spot by moving the cursor to the spot and pressing [F2]. This will anchor the upper left corner of the window when it loads.

## **Write WND**

To save a window, you have to tell the system what part of the visible screen is the window. Go to the left corner, press [Enter], then to the right corner, press [Enter]. Next you are asked to name the file. The WND extension will be added by the system.

**NOTE:** *By requiring you to tell the system exactly where the window area is prior to the Save, we give you the ability to save partial windows if you want. This can be helpful if you want to move a part of the screen. Do this by saving the small window, then loading it in again at the spot you designate with [F2].*

---

## **Help**

The Help system is there to remind you what is available while you are working. These function keys are active while you are on the editing screen so you do not have to go back to the menu.

F1	Help window (any key to exit)
F2	Set Upper Left Corner for Save, Load or Border
F3	Cycle through Text Color
F4	Cycle through Background Color
F5	Cycle down through Graphic characters
F6	Cycle up through Graphic characters
F7	Border
F9	Load Window
F10	Save Window
ESC	Exit to DOS
Return or Enter	Select Current Graphic Character
Ctrl-Backspace	Clear Screen
Alt S	Toggle Status Line
Ctrl-Break	Exit

## Appendix F

# Keyboard Return Codes

Keyboard return codes are returned as strings with the values listed below. These Codes are attained through the use of String Variable Input type 6.

<b>Key</b>	<b>Unshifted</b>	<b>Shifted</b>	<b>Ctrl</b>	<b>Alt</b>
F1	0 59	0 84	0 94	0 104
F2	0 60	0 85	0 95	0 105
F3	0 61	0 86	0 96	0 106
F4	0 62	0 87	0 97	0 107
F5	0 63	0 88	0 98	0 108
F6	0 64	0 89	0 99	0 109
F7	0 65	0 90	0 100	0 110
F8	0 66	0 91	0 101	0 111
F9	0 67	0 92	0 102	0 112
F10	0 68	0 91	0 103	0 113
Home	0 71	55		0 174
End	0 79	49	0 117	0 182
PgUp	0 73	57	0 132	0 176
PgDn	0 81	51	0 118	0 184
Ins	0 82	48	0 165	0 185
Del	0 83	46	0 166	0 186
Esc	27	27	27	
BkSpc	8	8	127	
Tab	9	0 15		
Return	13	13	10	
A	97	65	1	0 30

## Keyboard Return Codes

---

Key	Unshifted	Shifted	Ctrl	Alt
B	98	66	2	0 48
C	99	67	3	0 46
D	100	68	4	0 32
E	101	69	5	0 18
F	102	70	6	0 33
G	103	71	7	0 34
H	104	72	8	0 35
I	105	73	9	0 23
J	106	74	10	0 36
K	107	75	11	0 37
L	108	76	12	0 38
M	109	77	13	0 50
N	110	78	14	0 49
O	111	79	15	0 24
P	112	80	16	0 25
Q	113	81	17	0 16
R	114	82	18	0 19
S	115	83	19	0 31
T	116	84	20	0 20
U	117	85	21	0 22
V	118	86	22	0 47
W	119	87	23	0 17
X	120	88	24	0 45
Y	121	89	25	0 21
Z	122	90	26	0 44
[	91	123	27	
\	92	124	28	
]	93	125	29	
'	96	126		
0	48	41		0 129
1	49	33		0 120
2	50	64	03	0 121
3	51	35		0 122

## Keyboard Return Codes

---

Key	Unshifted	Shifted	Ctrl	Alt
4	52	36		0 123
5	53	37		0 124
6	54	94	30	0 125
7	55	38		0 126
8	56	42		0 127
9	57	40		0 128
*	42	0 114		
+	43	43		
-	45	95	31	0 130
=	61	43		0 131
'	44	60		
/	47	63		
;	59	58		

## Appendix G

# Font Selections

### KST FONTS

ASCII0	Aa Bb Cc Dd Ee Ff
BLOCK10	Fla Bb Cc Dd Ee Ff
FUTURA10	Fla Bb Cc Dd Ee Ff
HELV13	Aa Bb Cc Dd Ee Ff
HELV15	Aa Bb Cc Dd Ee Ff
ASCII5	Aa Bb Cc Dd Ee Ff
ASCII5B	<b>Aa Bb Cc Dd Ee Ff</b>
VG15X10	Aa Bb Cc Dd Ee Ff
VG15X10B	<b>Aa Bb Cc Dd Ee Ff</b>
ASC20	Aa Bb Cc Dd Ee Ff
ASC20B	<b>Aa Bb Cc Dd Ee Ff</b>
BLOCK20	<b>Fla Bb Cc Dd Ee Ff</b>
COUR20	Aa Bb Cc Dd Ee Ff
COUR25	<b>Aa Bb Cc Dd Ee Ff</b>
FUTURA20	<b>Aa Bb Cc Dd Ee Ff</b>
HELV20	<b>Aa Bb Cc Dd Ee Ff</b>
HELV21	Aa Bb Cc Dd Ee Ff
HELV25	<b>Aa Bb Cc Dd Ee Ff</b>

**SERIF25**

TIMES20

TIMES25

***TIMES25I***

TYPE25

TYPE25B

***TYPE25I***

VG20X

**Aa Bb Cc Dd Ee Ff**

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd Ee Ff

***Aa Bb Cc Dd Ee Ff***

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd Ee Ff

***Aa Bb Cc Dd Ee Ff***

Aa Bb Cc Dd Ee Ff

**BLOCK30****BLOCK40****Aa Bb Cc Dd Ee****Aa Bb Cc Dd**

COUR30

COUR35

COUR37

COUR40

AaBbCcDdEeFf

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd Ee

Aa Bb Cc Dd Ee

FUTURA30 Aa Bb Cc Dd Ee Ff  
FUTURA40 Aa Bb Cc Dd E

**HELV30 Aa Bb Cc Dd Ee Ff**

HELV31 Aa Bb Cc Dd Ee Ff

**HELV40 Aa Bb Cc Dd**

TIMES30 Aa Bb Cc Dd Ee Ff

TIMES30B Aa Bb Cc Dd Ee Ff

**TIMES37B Aa Bb Cc Dd Ee**

TIMES40 Aa Bb Cc Dd Ee F

## VV FONTS

PC-ROM

System

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd Ee Ff

Aa Bb Cc Dd

Aa Bb Cc

Roman1

Roman2

Gothic1

Gothic2

Script1

Script2

Italic1

Italic2

## **Appendix H**

# **Technical Information**

This appendix contains information on certain technical aspects of TEMPRA Media Author in respect to your PC. The section on CONFIGMA is especially important. Should you need to call our Tech Support line (1-813-682-1130) being familiar with this section can make your call more efficient and effective.

### **CONFIG.SYS and the AUTOEXEC.BAT**

During installation, TEMPRA Media Author automatically updates and changes your AUTOEXEC.BAT and CONFIG.SYS if necessary. The following samples represent the most “vanilla” form of these two files.

#### **CONFIG.SYS**

```
BUFFERS=30
FILES=30
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE 2048 X=A000-BFFF D=64RAM
I=E000-EFFF
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\SETVER.EXE
```

- Buffers do not have to be set at "30"
- Devices loaded after "DOS=HIGH,UMB" should be loaded high, but is not required. The statement used to load devices high should begin "DEVICEHIGH"
- Loading devices "HIGH" will free up conventional memory. This is especially useful when driving audio boards with the Voyetra drivers.

### ***AUTOEXEC.BAT***

```
ECHO OFF
PROMPT $p$g
PATH C:\;C:\DOS;C:\TEMPRA;C:\CINE;
SET TMP=C:\
SET EDITOR=TEXTEDIT
SET PROGX=CINE
SET VVMODE=SGA_2E
SET VVLINES=200
SET VVPAINT=TEMPRA
SET BLAST_BUF=10
SET SPRITE_BUF=20
```

Several of these commands are placed into the AUTOEXEC.BAT during installation of TEMPRA Media Author. These include:

```
SET TMP
SET EDITOR
SET PROGX
SET VVMODE
SET VVLINES
SET VVPAINT
```

The remaining set commands are optional. They are used to allocate a ##K buffer size for audio (BLAST\_BUF) and for sprites (SPRITE\_BUF).

## ***Configuration Program***

---

The CONFIGMA.EXE program is used to change the contents of the MA.CFG file. The MA.CFG file contains the controls that determine how TEMPRA Media Author operates. These controls have defaults set by Mathematica, Inc. The defaults set by Mathematica, Inc. should operate on most computer systems. You have to change the settings in MA.CFG using CONFIGMA.EXE if:

1. You have a special device attached to your computer such as a touch screen, VCR, special display board, audio board or other device.
2. You add or change a device attached to your computer.
3. You want to change the colors used on the workform.
4. The current configuration produces a conflict and TEMPRA Media Author does not work properly or does not work at all.

TEMPRA Media Author reads the MA.CFG file when TEMPRA Media Author is run. The MA.CFG file tells TEMPRA Media Author the devices attached to your computer and any other specific information needed. If that information is incorrect, TEMPRA Media Author may have erratic results or may stop running completely. If TEMPRA Media Author does not run, you should check the contents of MA.CFG for conflicting information. For example, having two serial port devices set to use the same serial port will cause TEMPRA Media Author problems.

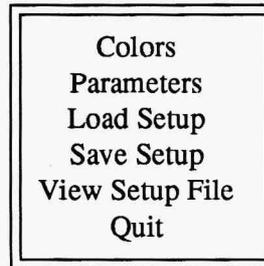
You may view the MA.CFG file by “typing” it to your screen or printing it. To print this file, be certain you are in the directory with TEMPRA Media Author. At the DOS prompt, type: **Print MA.CFG** and press [Enter].

When DOS asks the device name, press enter to use the standard printer (PRN). The printer prints a complete list of the contents of MA.CFG including all the values set.

The selections available in CONFIGMA.EXE may not be appropriate to all versions of TEMPRA Media Author. Some functions and devices are not supported but do have choices presented to you in CONFIGMA. These devices and settings were used in earlier versions or are being held for later versions.

To start CONFIGMA.EXE, be certain you are in the directory containing TEMPRA Media Author (\TEMPRA) and type: **CONFIGMA** and press [Enter]

The opening menu of CONFIGMA.EXE appears on your screen. The six choices available are:



You may move through the list by using the up and down arrow keys or by using a mouse. If you use a mouse, the left button acts as if it were the Enter key and the right mouse button acts as if it were the Escape key. Selecting Colors allows you to change the colors on your workform. Parameters changes the parameters and settings used by TEMPRA Media Author when it first starts and while it runs. Load Setup loads the copy of MA.CFG that is on your hard disk and Save Setup saves it back to your hard disk. The View Setup File option displays the MA.CFG on your screen. Quit stops the CONFIGMA program and returns you to DOS.

You can run the CONFIGMA program from within TEMPRA Media Author by selecting configure while in the workform menu. When you quit CONFIGMA you return to TEMPRA Media Author if that is where you started.

Don't do this to set an audio board. To set an audio board, run CONFIGMA from DOS before starting TEMPRA Media Author.

### **Colors**

The Colors selection leads to a menu where you can adjust the colors used in the Workform. You may not adjust the colors used on the Graphic User Interface (GUI). The colors that are used on the Workform when TEMPRA Media Author starts have been selected to provide an easily visible display. You can change the colors to suit your taste.

## ***Parameters***

The Parameters menu allows changes to the devices, addresses and controls used by TEMPRA Media Author. CONFIGMA.EXE presents the choices as a list. You may move through the list using the mouse or the cursor keys. When you find a setting you want to change, press enter or the right mouse button to see the options available to you. Some options provide for a choice and some options ask for a specific value to be entered.

### ***Joystick Param 1, 2 and 3***

If you have a Joystick rather than a mouse, you may adjust the sensitivity of the joystick. The parameters used are for the Kraft Joystick. The first parameter adjusts the wait between writing to the joystick while the second is the wait between reading the joystick port. The third parameter adjusts the wait between reading the joystick port and doing the next action. The range for each value is 0 to 255. The values must be typed into the box displayed.

### ***Videodisc Serial Port Com Port Parameter***

The four Serial Port Com choices are COM 1, COM 2, COM 3, or COM 4. The COM Port Parameter is for setting the Baud block for the serial communications: Baud Rate \ Parity \ Stop Bits \ # of Data Bits.

### ***Audio Board Type***

If your computer has an audio board, this parameter selects the type of board used. The choices are:

0 No Sound Board
6 Sound Blaster
7 MicroKey
9 Voyetra

Select the type of audio board in your computer by moving the highlighted bar to the correct board and pressing [Enter].

### ***TI Key Option***

For the Sound Blaster and sound boards that use Voyetra drivers you have a key option. This option will cause TEMPRA Media Author to stop playing

sound when a key is pressed. This is used only when the Play choice is used not Playasync.

### ***Touch Screen***

For systems that have a touch screen, this parameter allows the selection of the type of touch screen. Choose the screen you have attached to your computer.

<p>0 No Touch Screen 1 Sony PVM-1911 2 InfoWindow 3 MicroTouch 4 Tektouch 5 Elographics</p>
---

### ***Video Brightness, Hue, Contrast, Saturation***

These four parameters are used by the SNAPplus and the Video Blaster image capture boards. You may adjust the levels of brightness, hue, contrast and saturation for video as it enters the board. To see what they look like, use the capture feature from the TEMPRA PRO video menu.

### ***Video Input Standard (NTSC/PAL)***

This parameter allows you to adjust the type of video that TEMPRA Media Author expects to receive on an image capture card such as the SNAPplus. NTSC is the United States standard for video while PAL is the European standard. RGB input can have the Sync line be separate or on the green line. This is the difference between RGB\_SEP\_SYNC and RGB\_SYNC\_GREEN. Composite and Svideo are also separate video input methods that can be NTSC or PAL. The choices are:

<p>0x15 RGB_SEP_SYNC NTSC 0X25 RGB_SYNC_GREEN N TSC 0X35 COMPOSITE NTSC 0X45 S_VIDEO NTSC 0x16 RGB_SEP_SYNC PAL 0X26 RGB_SYNC_GREEN PAL 0X36 COMPOSITE PAL 0X46 S_VIDEO PAL</p>
---

### ***Audio INT/IMA/IRQ***

This parameter is intended for those who have an Audio board that has been moved to a non standard address or had other parameters changed. You may enter your own parameter or select one of the two provided parameters used with the MicroKey.

### ***Touch Screen Serial Port***

This parameter sets the serial port used by your touch screen. The choices start at COM1 and go to COM4 or no touch screen. You must set the COM port to where your touch screen is connected.

### ***Touch Screen Indicator***

This indicates the results of touching the screen. The choices are:

- |   |
|---|
| <ul style="list-style-type: none"><li>O - Just Process Touch</li><li>1 - Beep on Touch</li><li>2 - Highlight Touch</li><li>3 - Beep and Highlight Touch</li></ul> |
|---|

### ***VCR Type***

If your computer has a Sony VISCA controller, you can select this parameter. The choices are None or VISCA. If this parameter is set to on, TEMPRA Media Author can use a Sony VISCA controller.

### ***VCR Port***

You choose the port where the VISCA controller is located. The choices are Port 01 to Port 11. Until you save the setup, no changes made using the CONFIGMA program take effect. If you are unsure about an option or do not know specific information, you may quit CONFIGMA without making changes to the MA.CFG file.

## Contents of the MA.CFG File

```

0x70 [ menu_border_color ]
0x71 [ menu_text_color ]
0x4f [ menu_bar_color ]
0x07 [ request_text_color ]
0x07 [ request_border_color ]
0x70 [ request_highlight_color ]
0x71 [ form_highlight_color ]
0x17 [ form_norm_color ]
0x07 [ dir_border_color ]
0x07 [ dir_text_color ]
0x70 [ dir_bar_color ]
0x4f [ error_text_color ]
0x4e [ error_border_color ]
0x71 [ info_line_color ]
0x0e [ form_prev_color ]
0x17 [ form_color ]
0x17 [ help_border_color ]
0x17 [ help_text_color ]
    0 [ 0 = no joystick 5 = joystick ]
    0 [ 0 = no joystick 80 = joystick ]
    0 [ 0 = no joystick 150 = joystick ]
0x03f8 [ videodisc serial port address (hex) ]
0x0083 [ videodisc com port param (hex) ]
    6 [ audioboard 0=none 6=SnDBlast 7=MicroKey 9=Voetra]
    8 [ ti board gain setting]
    5 [ ti board volume setting]
    1 [ ti key option 0=abort any 1=enable func abort
nonfunc 2=enable fund 3=no abort ]
    0 [ 0=no touch screen 1=sony pvm-1911 2=infowindow
3=MicroTouch 4=TekTouch 5=Elographics]
0x03a8 [ PIB I/O base (hex) ]
0xd000 [ PIB address segment (hex) ]
    139 [ PIB brightness (dec) 0-255 ]
    131 [ PIB hue (dec) 0-255 ]
    125 [ PIB contrast (dec) 0-255 ]
    255 [ PIB saturation (dec) 0-255 ]
    2 [ WatsonII Com Port (1 or 2) ]
    1 [PIB + start U=underscan 0=overscan 0=NTSCO 1=NTSCU
2-PALO 3-PALU ]
    1000 [ EMS to Use (in K) ]
0x62 [ Wang Voice board software interrup - MicroKey
Audio INT/DMA/IRG ]
0x00 [ Videodisc Type ]
0x00 [ Touch Screen Com Port ]
0x00 [ Touch Screen Indicator ]

```

## ***Start up Options***

---

You can start TEMPRA Media Author with a number of options:

**/B for BIOS.** To speed up screen display of the workform we do not use BIOS. If there is a problem with going around BIOS in your system, start with the /B option and TEMPRA Media Author uses BIOS for text display in the 80 column mode.

**/C for Control C [^C].** When you run the RIV program, you can add this to the command line to allow the interruption of the sequence with ^C.

**/D to Disable Screen Clearing** on start up. If you have an image on the screen and you do not want to clear the screen, use this option when starting TEMPRA Media Author. This is really only applicable to the Frame grabber versions.

**/H or /? to Display These Flags.** If you want a little help with what options are available, at start-up use this option for help.

**/L for File Name** of the error log file. If you enter nothing here, the file is called MA.ERR.

**/R for Run.** This runs your sequence without bringing up the interface. Pressing [Esc] will stop the sequence run and the interface appears.

**/S to Disable Sound,** even though you may have it installed. If you have installed sound, but you do not want it to play, start with this option.

**/T to Calibrate the Touch Screen.** Use this option to calibrate the touch screen before TEMPRA Media Author runs.

**/V for Verbose.** If you are having trouble running TEMPRA Media Author in your computer, start the program with this option. Then you see initialization information displayed on the screen as it occurs in the computer. This is the same information you see in the MA.ERR file.

**/W for Workform.** You can start TEMPRA Media Author and see the Workform first if you start with the /W option.

**/Z for Trace Flag.** Use this to debug a sequence. This will give you a record of every line as it occurs.

### ***MIDI Event Information***

---

You will need to install Voyetra drivers (sold separately) to run a Midi event. Voyetra drivers let you choose any combination of the three types of drivers for your sound board. The first option is Midi and you would select the board you have from the list. The second option is internal FM sounds and the third is Digital Audio.

- If you have a MPU 401 compatible card with only Midi, you select None for options 2 and 3.
- If you have both FM and external Midi cards (SoundBlaster PRO, MediaVision Pro Audio Spectrum, etc.), you can use an external Midi device to play .MID files. You need to make sure that the Midi file was saved in your sequencer program with the port number you need. Using Sequencer Plus from Voyetra, which comes with the SoundBlaster Pro, Port 2 is internal FM synth and Port 1 is external Midi (keyboard, etc.).
- When Voyetra drivers are installed you will have a VMP subdirectory. With a text editor, you need to add two statements to your VMP statement

`C:\VMP\VMP /CLP:XX /mport`

`\CLP:XX` The XX is the size, in Kilobytes of the largest MID file that you will be using. (i.e., CLP:30 is 30k)

`\mport` Instructs VMP to load in Port Mode. Normally VMP defaults to MME Mode.

You will need to run the file DRIVER.BAT from DOS before you start TEMPRA Media Author.

- Before you start TEMPRA Media Author, you need to run CONFIGMA.EXE, which is located in the same directory as TEMPRA Media Author. Select Parameters and then choose Audio board type. Select (9) Voyetra, then press [Esc], then select QUIT, and then SAVE. If you type MA -V to enter TEMPRA Media Author, you will see a display and text of all the equipment that you selected in the CONFIGMA file.
- Make sure that your MID files are in the default subdirectory, TEMPRA\AUD, or that you change your path for Audio files using the PATH button on the Menu Bar.

## ***Touch Screen Information***

---

Interaction through a touch screen is very satisfying for users because it is as close as they can get to reaching into the screen and getting their wishes.

Naturally, a multimedia author will want to use a touch screen for such applications as kiosks, computer-based training, on-line reference, and device control. This section covers the material you need to know to succeed with touch screens and TEMPRA Media Author.

### ***Hardware***

The first step is making sure you have a touch screen that is compatible with the software. TEMPRA Media Author works with the following touch screens:

#### **TekTouch**

Graphics Technology Company

5524 Bee Caves Rd., Austin, TX 78746

512 328-9284 FAX, 800 962-3860, FAX in Texas 512 328-9285

#### **MicroTouch**

MicroTouch Systems, Inc.

55 Jonspin Rd., Wilmington, MA 01887

508 694-9900, FAX 508 694-9980

Elographics  
Elographics Inc.  
105 Randolph Rd., Oak Ridge, TN 37830  
615 482-4100

Other touch screens can be added for a fee.

### **Set Up**

These touch screens are generally connected to the computer via a serial RS-232 connection. Some have their own card, but we suggest the RS-232 connection for use with TEMPRA Media Author. Once you have the touch screen, connect it to a serial port that is different from the serial port you are using for your mouse.

Next make sure the touch screen itself is set to the following communications parameters. Check the manual for your touch screen to see how this is done.

Microtouch: 4800 or 9600 baud; 7 bits data, 2 stop bits, no parity

---

***NOTE:** When you are using the Microtouch you can use either 4800 or 9600 baud. Choose one for your system and set an environment variable VVMTBAUD= \_\_00. This tells TEMPRA Media Author what the baud rate is.*

---

TekTouch - 9600 baud, 7 bits data, 1 stop bit, even parity

Sony - 4800 baud, 8 bits data, 1 stop bit, no parity

Elographics - Automatic

Now that your system is set, you have to configure TEMPRA Media Author to read the touch screen.

- 1) Start CONFIGMA.
- 2) Select Parameters.
- 3) Select Touch Screen.
- 4) Choose your touch screen from the list.
- 5) Select Touch Screen Serial Port.
- 6) Select the port you plugged the RS-232 connector, or the port address of the card if the touch screen uses one.
- 7) Save the configuration.
- 8) Quit the program.

---

***NOTE:** You have the option of getting TEMPRA Media Author to give indications that the screen has been touched. You can set the touch screen to beep at a touch, highlight with the touch, or both. Use the Touch Screen Indicator setting in CONFIGMA for this.*

---

### **Starting TEMPRA Media Author**

Next you start TEMPRA Media Author. You will want to calibrate the touch screen if this is the first time you have used it, or you have moved the system. To do this, start TEMPRA Media Author by typing “MA -t”. Follow the screen prompts to calibrate the screen as TEMPRA Media Author initializes.

---

***NOTE:** If you test a touch screen event with a button event, and it is not working this may be due to port conflicts. Be sure to look through the CONFIGMA parameters to insure that no other serial devices are also set to your touch screen port. Next check to see that the hardware is set properly as outlined above.*

---

### **Working with Touch Screens**

The events that take touch screen input are: Button, Wait, and Timeouts. This is to say touch screens do not work with the TEMPRA Media Author interface (text or GUI), Dir events, or drawing commands.

### ***Buttons***

When you make a Button event for use with a touch screen you can use the default cursor to make the touch areas, or the touch screen.

To enable the mouse press [Alt] T, you will see a notice on the screen that the mouse is enabled. Press [Alt] T again and see the notice that the touch screen is enabled. Sometimes it is satisfying to use the touch screen as an author and set the button points with your finger.

When you are testing the button event with the touch screen use [F6].

### ***Wait For***

Use the touch screen as an “Any Key” in the Wait of an event.

### ***Timeout***

A touch-screen touch counts as a key press in Timeout events, so if you touch the screen, the time out is effected.

### ***Design Considerations***

- 1) Don't make the buttons too small, or close together.
- 2) Be sure that something the user can see happens right away so he knows he has pressed the screen and something happened. If you don't do this, you will find that the users will be pounding on the monitors and that will make no one happy.
- 3) Because people will be touching the screen, the areas they touch may become dirty causing unwanted glare and lack of clarity for the user who wants to enjoy your images and message. To minimize this problem, you may want to keep the touch points in one area of the screen away from the part of the screen where you are showing your information.

Once you have gotten familiar with using your touch screen with TEMPRA Media Author, you will find that it is an excellent interface for your users, and one that even the most anti-technology people will utilize.

## ***Environmental Variables***

When you installed TEMPRA Media Author, you were asked if you wanted to change your AUTOEXEC.BAT and CONFIG.SYS files. If you did, your path statement was changed to include the TEMPRA Media Author directory. TEMPRA Media Author also added some SET statements to your AUTOEXEC.BAT files. These SET statements create DOS environment variables. TEMPRA Media Author uses the DOS environment variables to establish its configuration every time you start TEMPRA Media Author. If you want to change any of these variables, use the following parameter guidelines. Check your DOS manual for information on setting environmental variables. The following describe the variables you can set for TEMPRA Media Author. To set a specific variable, enter the value after the = sign.

### ***BEEP***

**Sets frequency and duration of computer beep.**

(SET BEEP = 100 100) This variable sets the frequency and duration of the computer beep you hear if you ever make a selection the system cannot accept. If you want to set this (optional), you must have both entries. SET BEEP=100 100 sets the beep at 100 for 100 thousandths of a second. If you set only one number, the system hangs up.

### ***BLAST\_BUF***

**Sets a buffer for audio when using a Sound Blaster card.**

(SET BLAST\_BUF = ##) Valid entries, between 2 and 32, signify the size in kilobytes of the buffer. This allows audio to be played from the buffer (in the computer's memory) rather than from the disk, which would not be fast enough in some circumstances. If you have a large Blast\_Buf, you may not have room in memory for certain video modes, and the system uses the disk memory manager. This slows down execution.

### ***EDITOR***

**Sets the text editor to run from TEMPRA SHOW.**

(SET EDITOR = Editor Name) This determines the word processor used when you select Text Editor from the Workform or Word from the graphics

Menu Bar. If you don't set an editor, TEMPRA Media Author goes to the default text editor.

### ***MIDI\_BUFFER***

(SET MIDI\_BUFFER = ##) Valid entries are 2 through 64, expressed in K.

### ***MTBAUD***

Sets baud rates for MicroTouch touch screen.

(SET MTBAUD=####) Valid numbers: 9600 and 4800. This sets the baud rate (unit of speed) for MicroTouch Touch Screen's serial communications.

### ***PROGX***

Sets the external program for access from TEMPRA Media Author..

(SET PROGX=Program name) This lets you access any external program from the TEMPRA Media Author Menu Bar. You can set any frequently-run program for use without leaving TEMPRA Media Author. Return to TEMPRA Media Author by typing EXIT and pressing [Enter].

### ***TMP***

Tells which directory disk memory resident information is stored.

(SET TMP= Drive\Directory name) This determines in which directory on your disk memory resident information is stored. This TMP file also contains screen information for the Super VGA modes. This has to be stored in a file with a unique file name that TEMPRA Media Author makes and saves temporarily on the hard disk when you have a 640K machine.

### ***VDISC\_BAUD***

(VDISC\_BAUD = XXXX) The default is 9600. You can set it to a whatever your player needs.

### ***VMP\_FORMAT***

(VMP\_FORMAT = VOC) This allows you to use VOC files instead of WAV files with Voyetra drivers.

**VVDAC**

(SET VV\_DAC = VGA) Certain video boards without a high color DAC or an EDSUN DAC do not allow the program to start unless this is set.

**VVLINES**

**Determines the number of lines in a TEMPRA Media Author sequence.**  
(SET VVLINES = ##) Valid numbers are between 50 and 500. You can set the number of lines to suit your computer's memory. Use the same number when editing and running your sequences. If the number of lines in a sequence is more than the number of lines allowed with this setting, extra lines are cut off the end of the sequence.

**VVMODE**

**Sets the starting graphics mode and color depth.**

(SET VVMODE=Mode) To set the 320x200 graphics mode with 256 colors, use the format SET VVMODE=VGA\_13. TEMPRA Media Author selects a mode that most closely matches your display if you don't set one. The installation automatically sets this for you and puts this into your AUTOEXEC.BAT.

**VVPAINT**

**Sets TEMPRA PRO or TEMPRA GIF as external paint program.**

(SET VVPAINT=Program name) This determines what paint program is executed when you select the "TEMPRA" button on the Menu Bar. TEMPRA uses .PCX, .TGA, .WIN, and .PCC files. If you don't set this, you can't run a paint program from the "TEMPRA" button on the Menu Bar.

**VVPIC**

(SET VV\_PIC = TGA) Use this to set the kind of image format you intend to use. Your options are TGA for 32,768 color 16-bit TGA format images, PCX, and CEG for Edsun Continuous Edge Graphics.

**VVSHARE**

(SET VV\_SHARE = SHARE) Use this when you want to allow file sharing over a network.

### ***VVSPRITE\_BUF***

**Sets the memory storage area size for sprites (picture clips).**

(SET VVSPRITE\_BUF=##) Valid numbers are between 2 and 64 (expressed in K). If you have EMS you don't need this. You can vary this to accommodate memory constraints in your PC. A large VVSprite\_Buf allows you to show large image clips, but this large buffer may preclude you from using a number of VGA modes in conventional memory.

### ***VV\_VARK***

(SET VV\_VARK = ##) Use this when you want to set aside a large buffer for variables. Valid numbers are between 8 and 63 and represent kilobytes.

## ***Troubleshooting TEMPRA SHOW\_\_\_\_\_***

If TEMPRA Media Author does not start properly, there are a number of actions that can be taken to correct the problem. TEMPRA Media Author should start the first time, display the copyright information, and wait for the user to press [Enter]. When that does not happen, the following should allow you to find the problem and fix it:

1. Check the amount of memory available. TEMPRA Media Author requires about 500Kb of memory in the standard DOS memory area. A total of 525Kb is preferred. TEMPRA Media Author will also use EMS or XMS memory. If you have a 286 class computer, you should add a LIM type memory manager. On 386 and higher level processors, XMS will be adequate. Check for TSR programs that are not being used, device drivers and other programs that use memory in the 0-640K area. TEMPRA Media Author does not need the Smartdrive device to run. If you are still low on memory, you can cut down on the amount of memory used in TEMPRA Media Author by restricting the number of lines used in each sequence. The default number is 500 lines. The number of lines is limited by a set statement called "VV LINES ." To change to fewer lines, add a statement to your AUTOEXEC.BAT file that looks like this:

SET VV LINES=100

This will restrict the number of lines to 100. Cutting down on the number of lines will cut down the amount of memory needed by TEMPRA Media Author.

2. Be certain the type of display specified during the installation of TEMPRA Media Author is the correct display. This variable is set in AUTOEXEC.BAT and looks like this:

```
SET VVMODE=VGA 13
```

If this is set incorrectly, TEMPRA Media Author does not work properly. To test this, change the AUTOEXEC.BAT file so that there is no 'SET VVMODE' statement. An alternative way of testing is to type 'SET VVMODE=' at the DOS prompt and then run TEMPRA Media Author. If TEMPRA Media Author runs, the problem is the set statement. To find the proper set statement, try different video modes from within TEMPRA Media Author. Create a set statement in your AUTOEXEC.BAT with the video mode that works best for you.

3. The TEMPRA Media Author set up program CONFIGMA.EXE has settings in it that conflict with your computer. CONFIGMA.EXE changes the file called MA.CFG. When TEMPRA Media Author starts, it reads MA.CFG and acts accordingly. If you have changed your settings using CONFIGMA.EXE or the default setting provided with TEMPRA Media Author are not correct for your computer, TEMPRA Media Author will not run. To test this error, rename the file MA.CFG to a file called XMA.CFG and start TEMPRA Media Author. If TEMPRA Media Author runs, there is a problem with the settings in the MA.CFG program. To cure the problem, rename the XMA.CFG file to MA.CFG and run the program CONFIGMA.EXE. In the CONFIGMA.EXE program, turn off all the devices that you may not need such as touch screens, PC-VCR or other devices. If the three steps listed above do not solve the problem, you may need to contact Mathematica, Inc. Technical support.

4. To supply compatibility with all the latest video graphics boards, even the ones that were not built when the software was finished, TEMPRA Media Author works with VESA drivers. These DOS loadable drivers allow us to work with all boards through industry-standard software. If TEMPRA Media Author does not run your board at all, set the Environmental Variable `VVDAK = VGA` and load the VESA driver for your board before starting TEMPRA Media Author.

### ***More Troubleshooting***

---

In the process of making sequences you may get an occasional error message. These messages are stored in a file called MA.ERR that is made while TEMPRA Media Author is running. Please have that file available when you call us. It will help us to solve your problem. Many of the problems and inconsistencies you are likely to see are a result of insufficient memory.

If you are getting “**File too large**” messages when you are trying to load Sprites, you are out of memory and do not have enough room in your PC for the Sprite. It is very hard to tell if your Sprite file is too large by looking at the size of the file as it is saved on the disk. This is because the Sprites are saved as compressed images, and they decompress after they load into memory. The rule of thumb here is to make the Sprites for 256 color images no bigger than the 320x300 256 color screen.

**The Text Event Width/Height screen does not clear to reveal your graphics screen.** This is due to lack of free memory. You are in a mode that takes up too much memory to allow TEMPRA Media Author to keep a good copy of your graphics screen in memory.

If you get the message “**Error creating mask for \*.PCC**” this is due to the video mode. The Sprite you selected from the Sprite directory menu is made in a mode that is not the same as the current video mode.

If you are using TEMPRA Media Author without EMS, you can get more memory space by reducing the number of files and buffers in your CONFIG.SYS file. We recommend not going below 10 files and 10

buffers. Also set your VVLINES and SPRITE BUF and TMP Environmental variables to maximize the system. Also use few if any Permanent events. Remember that **Super VGA modes will not work quickly without EMS.**

**Funny or “solarized” colors on the graphics screen.** The VGA system uses color palettes to show the best 256 colors for your image. All images do not have the same colors. When the colors change from screen to screen, it becomes necessary to provide the ability to load in a **new set of 256 colors.** This set of colors is a palette. If you load a palette for image 2 while image 1 is still on the screen, the colors of image 1 will go weird. To avoid this, show a color background (BColor), then load in the new palette. Finally load in image 2.

**Palette loading does not work.** If you have an early version of PC Paint, or PC Paintbrush, the PCX images were not saved with their palette information. The solution is to go back to your old version of the paint program, and load in your image. When it looks right, use a new screen grab program to save the image with the palette in it.

**Installing EMS.** If you do have EMS please do not install it at A000, because that is where your VGA card is.

**Can’t find the file you want.** Make sure the file is in the directory that the \*.PTH file is telling TEMPRA Media Author to look in. Go to the Path menu and modify the path to the directory where your file is stored so TEMPRA Media Author can find it.

**Things don’t line up right** when using a ViewPort. This is due to a byte boundary problem. If you are having this problem, position both sides of your ViewPort on a byte boundary. To do this you have to use the following rule: If you are not in a 256-color mode, the x coordinates of the ViewPort have to be divisible by 8.

**Your paint program does not make PCC files.** If you have saved a picture clip to be used as a Sprite in TEMPRA Media Author, and it has been saved as a PCX file, you have to rename it as a PCC file. Do this with the following

DOS command: "RENAME {filename}.PCX {filename}.PCC." For example: RENAME FLOWER.PCX FLOWER.PCC. TEMPRA PRO and TEMPRA GIF save \*.PCC files

**Your paint program does not make PCC files, but it does make CLP files.** If you are using PCPaint, your paint program will make CLP files when you save a clipping. To get TEMPRA Media Author to recognize these files, you must set an environmental variable. This variable is called VVPAINT. Add the following line to your AUTOEXEC.BAT file: SET VVPAINT = PCPaint. When you restart your system, TEMPRA Media Author will use your CLP files.

You are getting **Unexpected end of file** on your path files (\*.PTH) or the configure (\*.CFG) file. This happens as a result of modifications being made to these files with a text editor (word processor) that were not correct. Do not use a word processor to edit \*.CFG files.

**Error messages are just numbers or don't look like English.** This is probably a result of corruption of the MESSAGE.W file.

**EDITOR does not work.** There are a number of possible causes. Your EDITOR environmental variable is not set. Set it by adding the following line to your AUTOEXEC.BAT file:

SET EDITOR = {editor name}.

You must also have the editor where you can find it, either by setting the DOS path command, or by putting it into the same directory where you have TEMPRA Media Author.

Another possible cause is that there is not enough room in memory for TEMPRA Media Author and the Editor. If this is the case you will see the DOS message: Program too big for memory.

**If you have a digitizing tablet and TEMPRA Media Author does not work,** this may be due to conflicts between the table driver and the mouse

driver TEMPRA Media Author is expecting. If you have an MSmouse driver for the tablet, use that and you can drive TEMPRA Media Author with your stylus.

**Super VGA modes do not work.** We do not support all super VGA boards, only those with the following VGA chips: Tseng Labs, Trident, Paradise, and Video 7. If you have a board made with one of these chips, you will still need to have EMS to use the super VGA modes. We will continue to update the system.

**The System is slow.** Without EMS, TEMPRA Media Author will not be fast in Sprite movements or wipes. To speed up the wipes you can set the Global Method Quick ON. This speeds up the wipes by only using the Quick wipe. Be aware of the memory constraints and optimize your system with the environmental variables. If you have EMS and the system is slow, this could be because you have too many Erasing or Permanent Events; use Temporary Events when possible.

**You can't see anything on your monochrome monitor.** Please use the CONFIGMA program to set the workform colors so you can see the fields, cursor, menus, and highlight. We would like to encourage you to get a color monitor, so you can see the full benefit of color graphics as TEMPRA Media Author shows them.

**System will not start.** Be sure you have set your TMP environmental variable. This will let you run TEMPRA Media Author in all modes even without EMS.

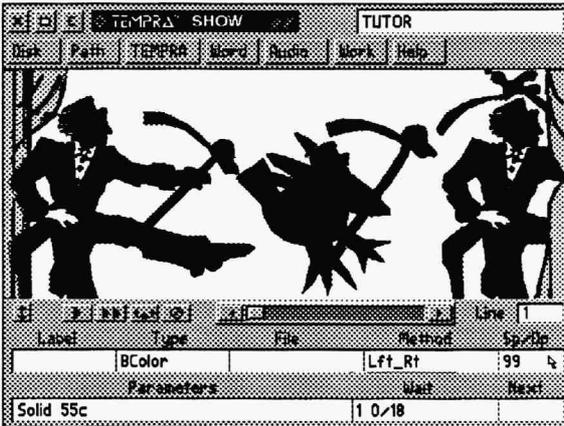
**You are not getting prompts or error messages you can understand.** Go to your original diskettes and reinstall the program.

**SoundBlaster clicks as audio is playing.** Enlarge the BLAST\_BUF.

**SoundBlaster is playing and the system dies.** You are trying to play a new audio file while another file is playing asynch. If you are using Voyetra drivers, reduce the size of the Voyetra buffer. Be sure to contact Voyetra about compatibility with their software and your hardware configuration.

**Out of Environment Space.** Be sure your Shell command ends with the \E:1024 option, for example

Shell = C:\COMMAND.COMC:\VP/E:1024.



## Appendix I

# TEMPRA SHOW

## Tutorial

### Introduction

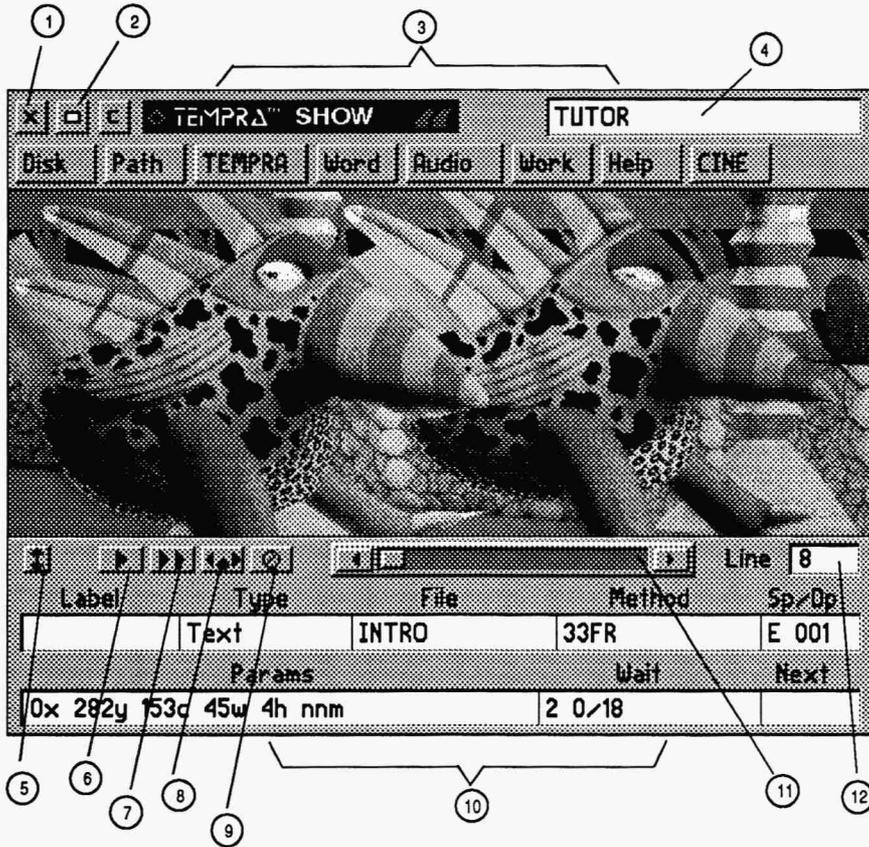
This hands-on walk-through tutorial gives you an opportunity to explore TEMPRA Media Author. While this tutorial was written for TEMPRA SHOW, you may use it to learn about events available in TEMPRA Media Author. Another tutorial in Appendix J demonstrates the more advanced features of TEMPRA Media Author. The written tutorial we've designed lets you recreate an interactive sequence included with this program (called TUTOR.SEQ). If you have already read through the user guide, you will be familiar with many of the terms used in this tutorial.

We suggest you read each page carefully, noting the **bold-faced** words, as they are key terms that relate to a specific event or action. There are 24 lines in the sequence you will recreate. If you're familiar with computers, you'll breeze through the tutorial. If you're new to computers, it may take a little longer to complete the tutorial, but you will get faster as you become familiar with the process. It's exciting to watch the sequence as you create it and when you're finished. This sequence was designed for 640x480, 256-color display modes. If you don't have this display mode, your results will be different.

---

**NOTE:** *If you press the wrong key for a specific line or notice that the menus don't match the written material, we suggest you cancel out of that line, delete it, and redo the line from the beginning. This will help keep you in step with the tutorial.*

---



- |   |                    |    |                      |
|---|--------------------|----|----------------------|
| 1 | Exit Button        | 7  | Play Sequence Button |
| 2 | Remove Menu Button | 8  | Insert Line Button   |
| 3 | Menu Bar           | 9  | Delete Line Button   |
| 4 | Title Box          | 10 | Event Controller     |
| 5 | Menu Size Button   | 11 | Line Scroll Bar      |
| 6 | Play Line Button   | 12 | Line Counter         |

## ***Using the Tutorial***

This hands-on walk-through tutorial lets you explore TEMPRA Media Author. It explains the events in a show we created for TEMPRA SHOW, called TUTOR.SEQ, so you can replicate them in TEMPRA Media Author, yet **add your own creative flair**. The following key points are extracted from the user guide as a reminder:

- When you start each new line with either the **Label** or **Type** field, TEMPRA Media Author automatically brings up any menus you need to complete the event. If you don't need a particular menu, a message appears telling you so.
- Use the **[F6]** key to play the sequence without the menus on the screen. The sequence begins with the current line shown in the Line Counter. To start a sequence from the beginning, change the Line Counter to read Line 1.
- To stop playing a sequence, press the **[Esc]** key. The sequence freezes and the menus appear on-screen.
- **Select** means to press the left mouse button. **Cancel** means to press the right mouse button.

## ***Tutorial Structure***

Beginning on page 8, this tutorial consists of two sections: the left-hand page contains a graphic version of how to create each line. At the top of the left-hand page is an objective to describe what you will accomplish in that line. At the bottom of the left-hand page is a sample of the Workform (the text-based interface). The right-hand page has step-by-step instructions of how to create the line. When you're finished with the tutorial, you would have recreated the sequence you're about to watch!

---

**NOTE:** *As you becomes familiar with TEMPRA Media Author, you'll be able to tell the contents of each line just by looking at the graphic on the left-hand page.*

---

## ***Starting the Tutorial***

Before you begin replicating the tutorial sequence we created specifically for this user guide, you should play it. To do this:

1. Change to the DOS prompt by typing **CD \** and pressing [**Enter**] (you see **C:\>** on the screen).
2. Change to the TEMPRA directory by typing **CD TEMPRA** and [**Enter**] (you see **C:\TEMPRA** on the screen).
3. Start TEMPRA Media Author by typing **MA** and [**Enter**].
4. Press [**Enter**] when you see the copyright message. The TEMPRA Media Author graphic interface appears on the screen.
5. Using the mouse pointer, select the **Disk** button from the Menu Bar by pressing the **left** mouse button.
6. When the Disk menu opens, select the **Load Sequence** bar by pressing the **left** mouse button.
7. Select **TUTOR.SEQ** to load the sequence.
8. Press the [**F6**] key to start the tutorial sequence. The menus disappear and the sequence begins. Watch the sequence and participate in it. (This is an **interactive** sequence. You choose the correct answer to complete the sequence.) The sequence ends with an image of the TEMPRA product box.
9. When you're finished with the sequence, press **any key** to return to the menus.
10. You are at the end of the sequence (**Line 23**). To get to the first line of the sequence, place your cursor in the **Line Counter** and press **Select** (**left** mouse button). The Line Counter menu pops up on the screen and displays **Line 1**. Press **Select** again to return to the first event in the sequence. The Line Counter displays **Line 1**.
11. Play with the **Line Scroll Bar** and **Line Counter**. Move forward and backward in the sequence by holding down the **left** mouse button as you move the control button inside the Line Scroll Bar. Also, try selecting the arrows on either side of the Line Scroll Bar. As you move back and forth, the number in the Line Counter changes. The information in the fields also changes.
12. To look at the **Workform** (text-based interface), select the **Work** button on the Menu Bar. The graphic menu disappears and the text menu appears on-screen. The screen shows all the events and characteristics. This information is the same as what is in the graphic menu, but in text form so you can see more of it at once.

13. Move the mouse around (but don't hold down any of the mouse buttons). The highlighted bar also moves around the fields.
14. Press **Cancel** on the mouse to exit the Workform and return to the graphic menu. The **Main Menu** pops up on the top left portion of the screen. Move the highlighted bar down to **GUI** and press **Select**. The graphic menu reappears on the screen.
15. To begin recreating the tutorial sequence, select the **Disk** button and then select the **New Sequence** bar. (If the message "Save old sequence?" pops up, select No. This shouldn't happen, but may if you change the original sequence somehow.)
16. The New Sequence menu appears. Type **your name** in the box and press **Select**. Your name appears in the **Title Box** on the Menu Bar. The Line Counter displays Line 1, and you're ready to start creating!

---

***NOTE:** Since you will have some creative license when you recreate the sequence, the numbers in your fields may not exactly match ours. But they will reflect the color, positioning, etc. that you have chosen.*

---

# TEMPRA SHOW: Line 1, BColor Event

*Objective: Wipe a color background onto the screen.*



Line 1 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	

Line: 1 Sequence: TUTOR F1-Help F5-Prev Line

## Line 1

Line 1 contains a colored **background** that wipes onto the screen from left to right.

---

*NOTE: You will not need to use the **Label** and **File** fields for this **BColor** type of event.*

---

### Recreating Line 1

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the Wipe menu, select **Lft\_Rt**. This wipes the color on the screen from left to right.
3. The **SP/DP** menu pops up. Press the Up arrow until the speed is **99**. Select **OK**. (Since all background color events are permanent, you can't choose from the **Display** field.)
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **Yes**. A cross-shaped cursor with a box around appears on the screen. Move the color pick cursor around on the screen. Notice how the box changes color to reflect the current color the cross is positioned over. Pick a color from the image and press **Select**.

---

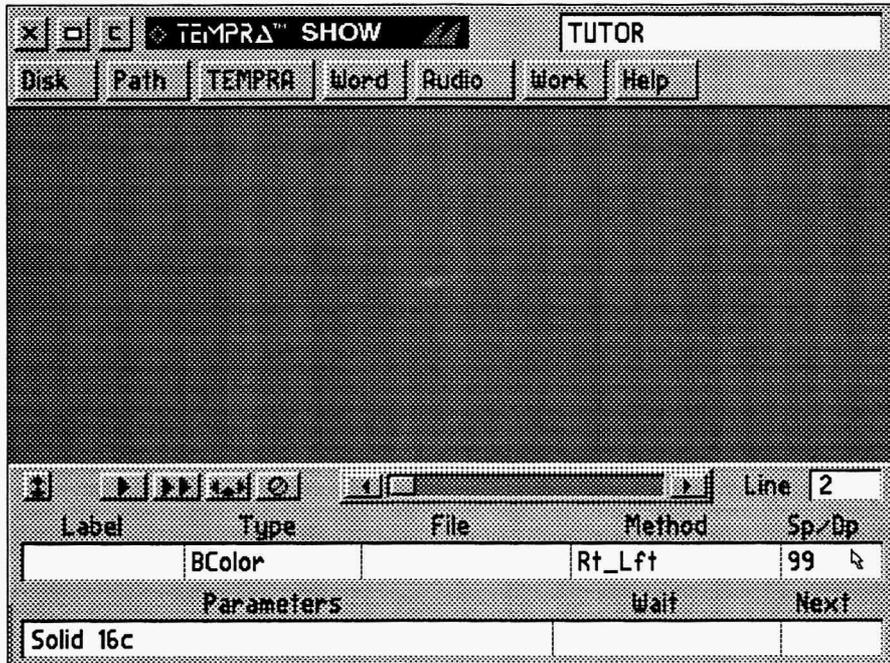
*NOTE: If you started this tutorial without first watching **TUTOR.SEQ**, you need to choose **No** on **Choose Color From Image?** This brings up the color palette for you to choose a color. See Line 2, step 4 for more information.*

---

5. The **Wait** menu automatically pops up. From the menu, select **Wait For Timeout**. The Seconds default is 1 (which we want), so press **Select**. Press **Select** again to accept the default on the 18th Seconds menu.
6. The **Next** menu pops up. Press **Select** to accept the default.
7. The screen color changes. The **Line Counter** advances to Line 2.

# TEMPRA SHOW: Line 2, BColor Event

*Objective: Wipe a color background onto the screen.*



Line 2 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		

Line: 2 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 2***

Line 2 contains a colored **background** that wipes onto the screen from right to left.

---

***NOTE:*** *You will not need to use the **Label** and **File** fields for this BColor event.*

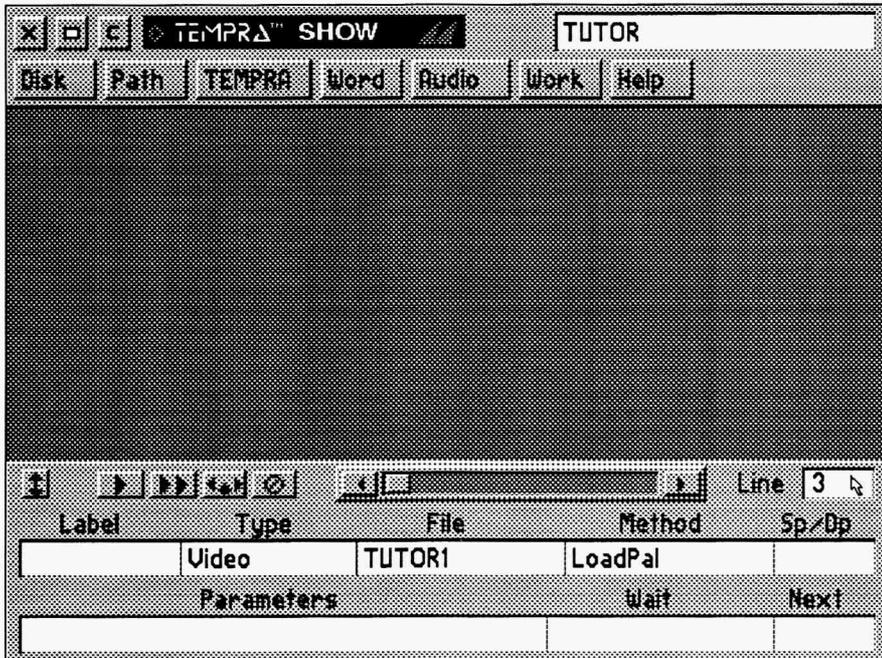
---

### **Recreating Line 2**

1. Select the box under **Type**. From the pop-up menu, select **BColor**.
2. The **Method** menu pops up. From the Wipe menu, select **Rt\_Lft**. This wipes the next color on the screen from right to left.
3. The **SP/DP** menu pops up. Press the Up arrow until the speed is **99**. Select **OK**.
4. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select any color square from the palette and select **OK**. (We used purple, number 16.)
5. The **Wait** menu automatically pops up. From the menu, select **Don't Wait**.
6. The **Next** menu automatically pops up. Press **Select** to accept the default.
7. The screen color changes. The **Line Counter** advances to Line 3.

# TEMPRA SHOW: Line 3, Video Event

*Objective: Load 256-color palette from an image.*



Line 3 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				

Line: 3 Sequence: TUTOR F1-Help F5-Prev Line

## Line 3

Line 1 contains a **Video** event. In this case, it loads in a 256-color palette from an image, called TUTOR1.PCX. The palette prevents the introductory screen from changing colors during the transition to the first image. If you don't load a palette based on an image you will use in the sequence, TEMPRA SHOW automatically loads the default palette.

---

*NOTE: You will not need to use the **Label**, **SP/DP**, and **Parameters** fields for this Video event.*

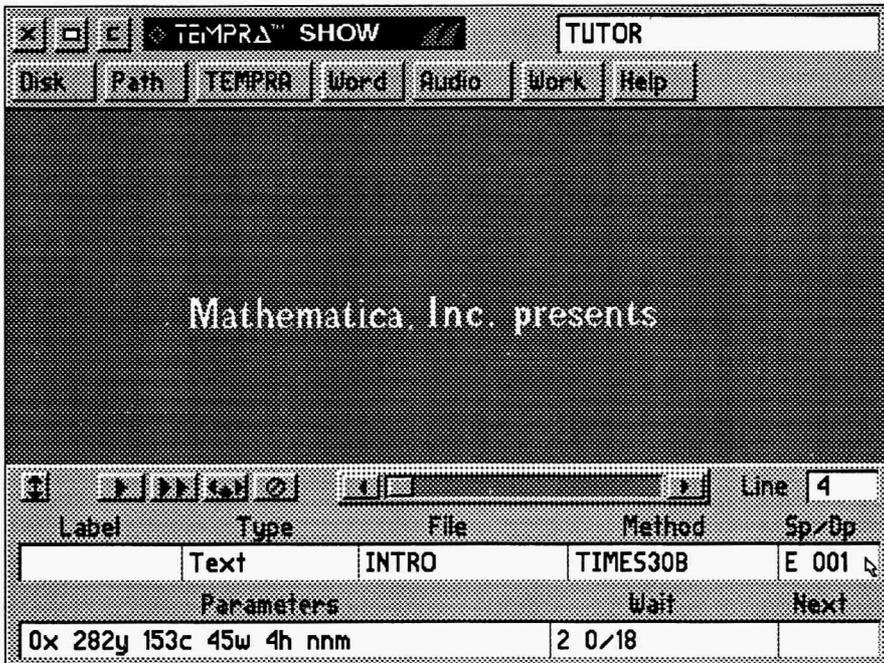
---

## Recreating Line 3

1. Select the box under **Type**. From the pop-up menu, scroll down and select **Video**.
2. The **Method** menu automatically pops up. Select **LoadPal**.
3. The **Files** menu automatically pops up. You will select an image file on which to base the 256-color palette. Since the first image we use in the sequence is TUTOR1.PCX, that should be the palette to load. Select **TUTOR1.PCX**.
4. The **Wait** menu automatically pops up. From the menu, select **Don't Wait**.
5. The **Next** menu automatically pops up. You don't need this, so press **Select** to accept the default.
6. The menus change color to reflect the new palette. The **Line Counter** automatically advances to Line 4.

# TEMPRA SHOW: Line 4, Text Event

*Objective: Place erasing text on the screen.*



Line 4 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nnm	2 0/18	

Line: 4 Sequence: TUTOR F1-Help F5-Prev Line

## Line 4

Line 4 contains a **text** file that we created in a word processor and saved as an ASCII text file.

---

*NOTE: You will not need to use the **Label** field.*

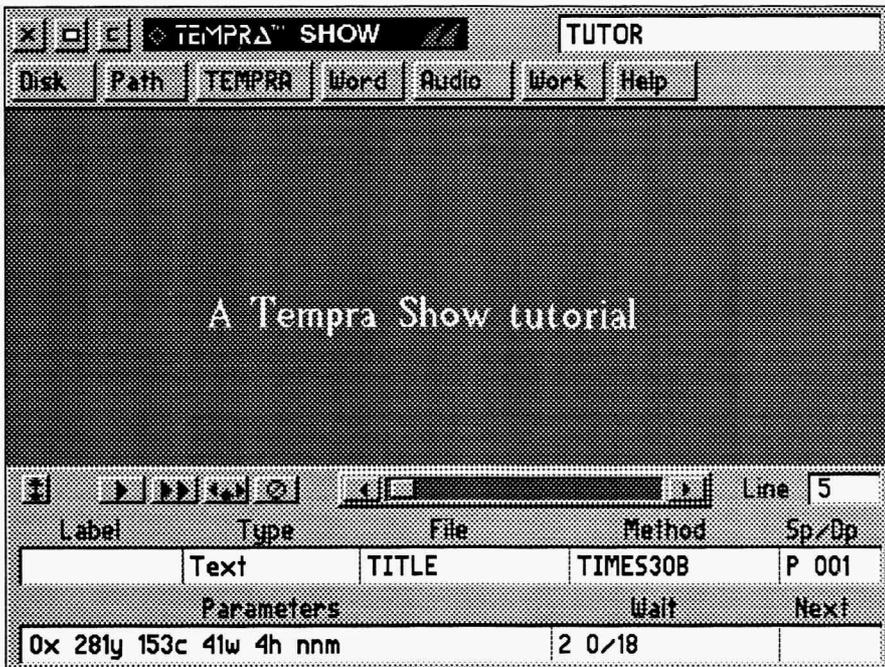
---

### Recreating Line 4

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **INTRO.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you'd like to use or scroll down and select **Times30** (Times Roman 30-point text).
4. The **SP/DP** menu pops up. Select **Erase after Displaying**. (We want Line 5's text to appear in the same place as Line 4's text.)
5. **Parameters** pops up. For **Choose Starting Line For Text**, select the first line in "Mathematica, Inc..." For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. Select a color square from the palette and select **OK**. (We used light blue, number 153.)
6. Framed text appears in the top left corner. **Move the frame around the text you want to display** (in this case, "Mathematica, Inc. presents) and press **Select**. The text appears on the screen to be positioned. Move the text to the **center of the screen** and press **Select**.
7. The **Wait** menu pops up. Select **Wait For Timeout**. Type **2** on Seconds menu. Press **Select**. Press **Select** again for the default of 0 on 18th Seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on-screen. The **Line Counter** advances to Line 5.

# TEMPRA SHOW: Line 5, Text Event

*Objective: Place permanent text on the screen.*



Line 5 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nnm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nnm	2 0/18	

Line: 5 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 5***

Line 5 contains **text** we already typed and saved as an ASCII text file.

---

***NOTE:** You will not need to use the **Label** field.*

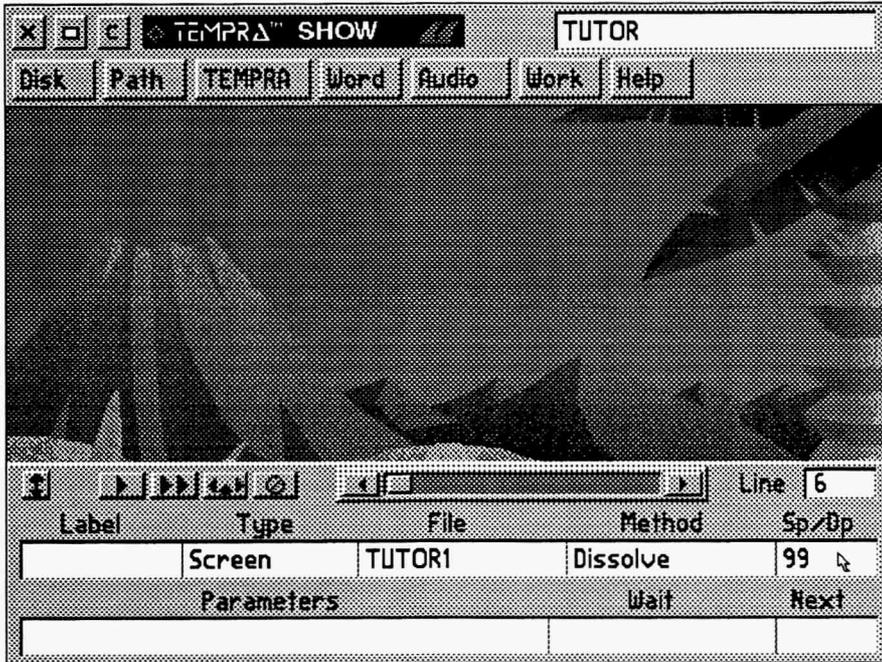
---

### **Recreating Line 5**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **TITLE.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you'd like to use or scroll down and select **Times30** (which we used).
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select the **first line** of "A Tempra Show tutorial." For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **No**. From the palette, select a color square (we used the light blue, number 153) and **OK**.
6. Framed text appears in the top left corner. **Move the frame around the text you want to display** (in this case, "A Tempra Show tutorial") and press **Select**. Move the text toward the **center** of the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait For Timeout**. Type **2** on Seconds menu. Press **Select**. **Select** again for default of 0 on 18th Seconds.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on-screen. The **Line Counter** automatically advances to Line 6.

# TEMPRA SHOW: Line 6, Screen Event

*Objective: Dissolve a .PCX image onto the screen.*



Line 6 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nrm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nrm	2 0/18	
	Screen	TUTOR1	Dissolve	99			

Line: 6 Sequence: TUTOR F1-Help F5-Prev Line

## **Line 6**

Line 6 contains a **.PCX graphics image** that is dissolved onto the screen. This image is a background graphic that sets the scene for Buzz, the TEMPRA toucan, to appear.

---

*NOTE: You will not need to use the **Label field**.*

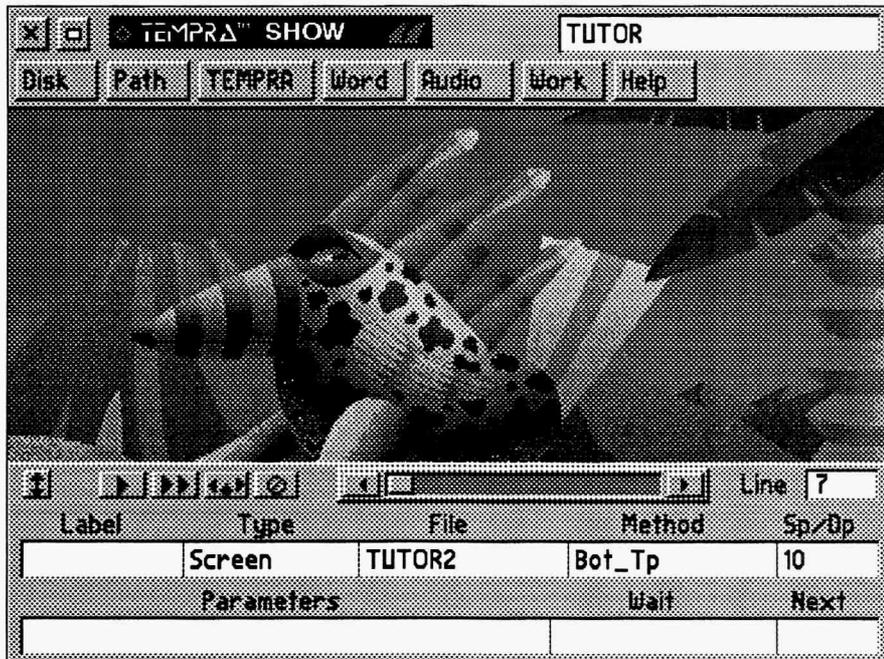
---

### **Recreating Line 6**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR1.PCX** from the list.
3. The **Method** menu pops up. From the Wipes menu, select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK** to accept the speed.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. The caveman background image appears on-screen. The **Line Counter** automatically advances to Line 7.

# TEMPRA SHOW: Line 7, Screen Event

*Objective: Wipe a .PCX image on the screen.*



Line 7 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Lft_Rt	99	Solid 55c	1 0/18	
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nmm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nmm	2 0/18	
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			

Line: 7 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 7***

Line 7 contains a **.PCX graphics image** of Buzz that wipes onto the background cave image.

---

***NOTE:** You will not need to use the **Label** field.*

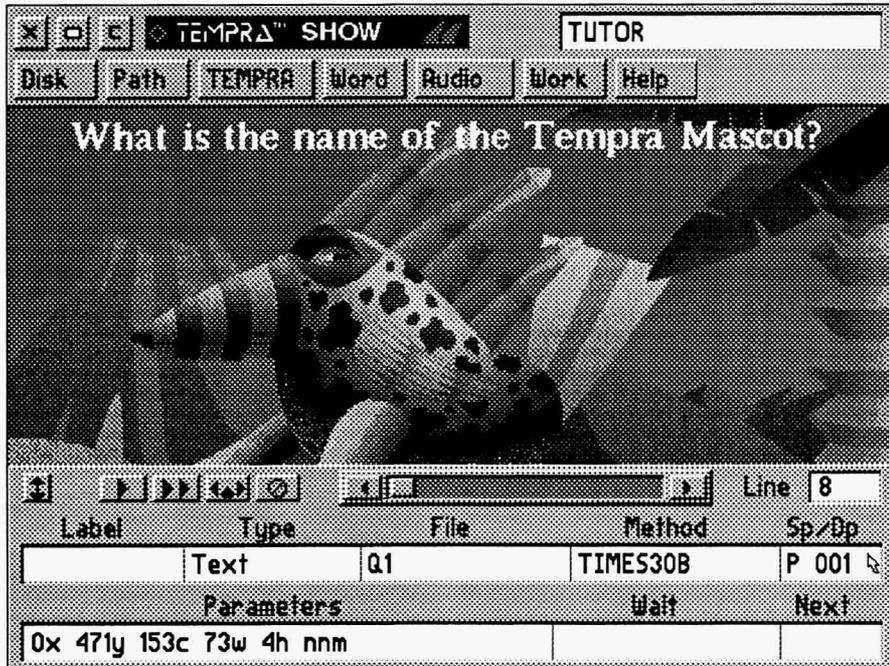
---

### **Recreating Line 7**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **TUTOR2.PCX** from the list.
3. The **Method** menu pops up. From the Wipe menu, select **Bot\_Tp**.
4. The **SP/DP** menu pops up. Set the speed at **10**. Select **OK** to accept the speed.
5. The **Parameters** menu pops up. For the **Show Screen** menu, we want the image to appear as it was created, so select **As Is**.
6. The **Wait** menu pops up. You want to continue to the next screen automatically, so select **Don't Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. Buzz wipes onto the caveman background on the screen. The **Line Counter** automatically advances to Line 8.

# TEMPRA SHOW: Line 8, Text Event

*Objective: Place permanent text on the screen.*



Line 8 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	BColor		Rt_Lft	99	Solid 16c		
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nnm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 28ly 153c 41w 4h nnm	2 0/18	
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 47ly 153c 73w 4h nnm		

Line: 8 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 8***

Line 8 contains the text “What is the name of the Tempra mascot?” You can treat this Text event the same as your earlier ones.

---

*NOTE: You will not need to use the Label field.*

---

### **Recreating Line 8**

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **Q1.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you’d like to use or scroll down and select **Times30** (which we used).
4. The **SP/DP** menu pops up. Select **Permanent**.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select on the first line in “What is the name of...” For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **No**. For **Justify**, select **Middle**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears on the screen. Move it around and select on the color you’d like the text to be.
6. Framed text appears in the top left corner. Move the frame **around the text you want to display** (in this case, “What is the name of the Tempra Mascot?”) and press **Select**. The text appears on the screen for positioning. Move the text **toward the tip of Buzz’s top blade** (that’s feather to you) so the text is visible even when the Menu Bar is on the screen and press **Select**.
7. The **Wait** menu pops up. Select **Don’t Wait**.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on-screen. The **Line Counter** automatically advances to Line 9.

# TEMPRA SHOW: Line 9, Sprite Event

*Objective: Move a sprite image across the screen.*

The screenshot shows the TEMPRA SHOW software interface. At the top, there is a title bar with 'TEMPRA SHOW' and 'TUTOR'. Below the title bar are several buttons: 'Disk', 'Path', 'TEMPRA', 'Word', 'Audio', 'Work', and 'Help'. The main display area shows the question: "What is the name of the Temptra Mascot?" with a background image of a spotted dog. Below the question is a control panel with various icons and a 'Line 9' indicator. At the bottom, there is a table for the current line.

Label	Type	File	Method	Sp/Dp
	Sprite	HAND		15 E
Parameters			Wait	Next
-55bx 288by 142ex 288ey			2 0/18	

Line 9 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Video	TUTOR1	LoadPal				
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nmm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nmm	2 0/18	
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nmm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	

Line: 9 Sequence: TUTOR F1-Help F5-Prev Line

## Line 9

Line 9 contains a **sprite** that points to Buzz. This hand sprite is also the image we use to branch later.

---

***NOTE:** You will not need to use the **Label** field.*

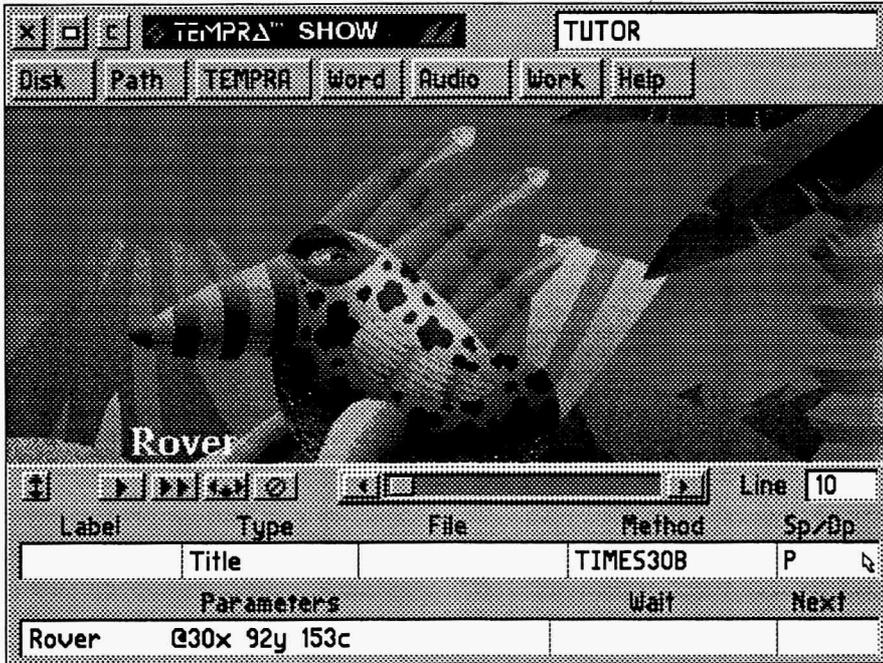
---

### Recreating Line 9

1. Select the box under **Type**. From the pop-up menu, select **Sprite**.
2. The **File** menu pops up. Select **HAND.PCC** from the list.
3. The **Method** menu pops up. From the menu, select **Simple** (for a basic sprite movement).
4. The **SP/DP** menu pops up. Set the speed at **15**. Select **Erase**. Select **OK**.
5. The **Parameters** menu pops up. Position the hand's **starting point** off-screen on the left side about eye level to Buzz, press **Select**. Then position the **ending point** near Buzz's eye, press **Select**.
6. The **Wait** menu pops up. Select **Wait For Timeout**. Type **2** on Seconds menu. Press **Select**. Press **Select** again for default of 0 on 18th Seconds.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. The hand moves to the right, pointing at Buzz. The **Line Counter** automatically advances to Line 10.

# TEMPRA SHOW: Line 10, Title Event

*Objective: Position a permanent title on the screen.*



Line 10 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Text	INTRO	Time30B	E 001	0x 282y 153c 45w 4h nrm	2 0/18	
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nrm	2 0/18	
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nrm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		Time30B	P 001	Rover @30x 92y 153c		

Line: 10 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 10***

Line 10 contains the **title** that is one of the three choices to answer the previous text question. You can create your own title or type the one we used.

---

***NOTE:** You will not need to use the **Label** or **File** fields.*

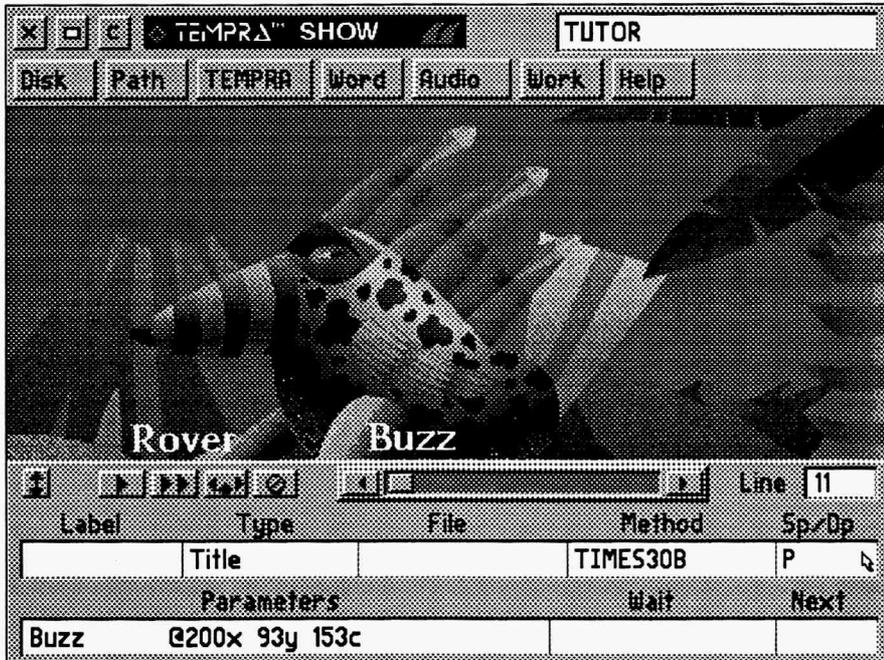
---

### **Recreating Line 10**

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you'd like to use or scroll down and select **Times30**.
3. The **SP/DP** menu pops up. Select **Permanent** and then **OK**.
4. **Parameters** pops up. In the **Input Title** box, type **Rover** (or make up your own name to use — a friend, boss, family, etc.). Press **Select**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears on the screen. Select a color from the image.
5. Text appears for positioning. Move the text **toward the bottom left of the screen** (above where the Event Controller would be so you can see the name when the Event Controller is on the screen) and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on-screen. The **Line Counter** automatically advances to Line 11.

# TEMPRA SHOW: Line 11, Title Event

*Objective: Position a permanent title on the screen.*



Line 11 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Text	TITLE	Time30B	E 001	0x 281y 153c 41w 4h nrm	2 0/18	
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nrm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		Time30B	P 001	Rover @30x 92y 153c		
	Title		Time30B	P 001	Buzz @200x 30y 153c		

Line: 11 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 11***

Line 11 contains the **title** “**Buzz.**” You can treat this Title event the same as the previous one.

---

***NOTE:*** *You will not need to use the Label and File fields.*

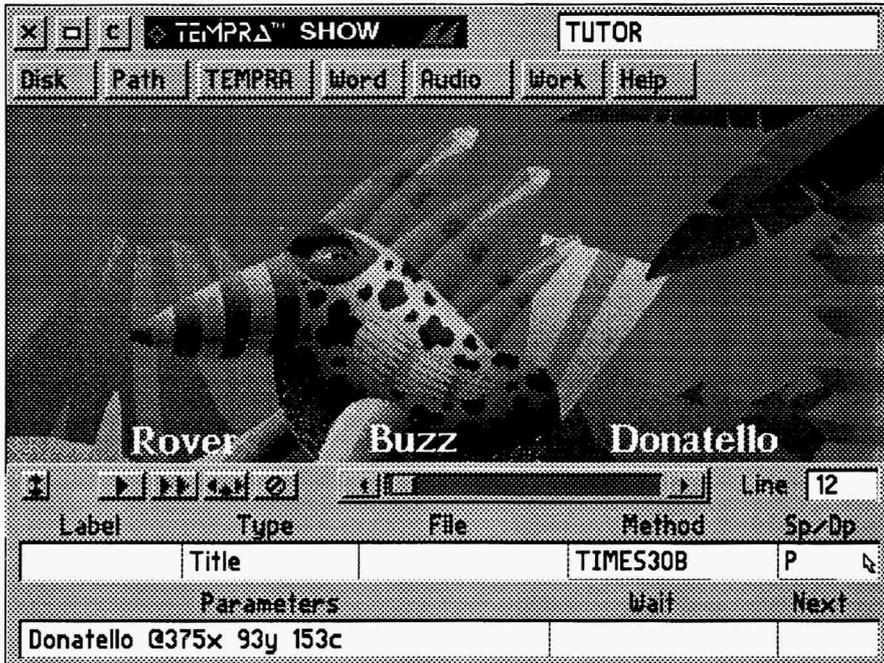
---

### **Recreating Line 11**

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you want to use or scroll down and select **Times30**.
3. The **SP/DP** menu pops up. Select **Permanent**.
4. **Parameters** pops up. In the **Input Title** box, type **Buzz** and press **Select**. For **Choose Color From Image**, select **Yes**. A cross-shaped cursor appears. Position over a color you want from the image and press **Select**.
5. Text appears on the screen for positioning. Move the text toward the **bottom middle of the screen** (about even with the first title) and press **Select**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu pops up. Press **Select** to accept the default.
8. Text appears on-screen. The **Line Counter** automatically advances to Line 12.

# TEMPRA SHOW: Line 12, Title Event

*Objective: Position a permanent title on the screen.*



Line 12 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Screen	TUTOR1	Dissolve	99			
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nrm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		Time30B	P 001	Rover @200x 92y 153c		
	Title		Time30B	P 001	Buzz @200x 80y 153c		
	<b>Text</b>		Time30B	P 001	Donatello @375x 93y 153c		

Line: 12 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 12***

Line 12 contains a **title** that is one of the three answers. You can treat this Title event the same as your earlier ones.

---

**NOTE:** *You will not need to use the **Label** and **File** fields.*

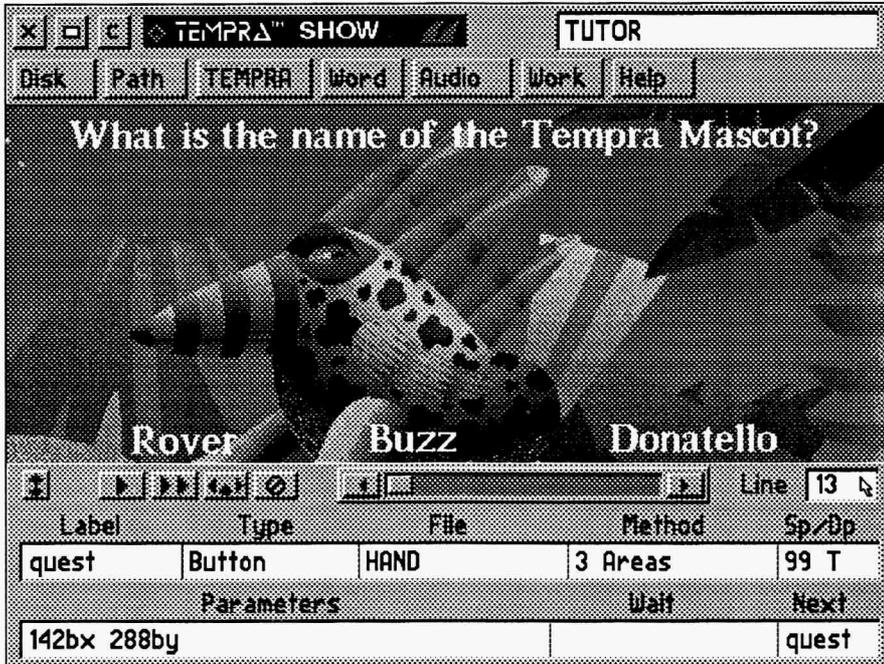
---

### **Recreating Line 12**

1. Select the box under **Type**. From the pop-up menu, select **Title**.
2. The **Method** menu pops up. Select **KST Fonts**. Scroll down and select **Times30**.
4. The **SP/DP** menu pops up. Select **Permanent** and then **OK**.
5. **Parameters** pops up. In the **Input Title** menu, type **Donatello** (or any other name you want to be the wrong answer) and press **Select**. For **Choose Color From Image**, select **Yes**. Using the cross-shaped cursor, select a color from the image.
6. Text appears on the screen for positioning. Move the text **toward the bottom right of the screen** (about even with the other choices) and press **Select**.
7. The **Wait** menu pops up. Select **Don't Wait**.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears on-screen (it may be hidden by the Event Controller). The **Line Counter** automatically advances to Line 13.

# TEMPRA SHOW: Line 13, Button Event

*Objective: Create a branch for interactive selection.*



Line 13 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Screen	TUTOR2	Bot_Tp	10			
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nrm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		Time30B	P 001	Rover @30x 93y 153c		
	Title		Time30B	P 001	Buzz @200x 93y 153c		
	Title		Time30B	P 001	Donatello @375x 93y 153c		
quest	Button	HAND	3 Areas	99 T	142bx 288by		quest

Line: 13 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 13 (through 17)***

Line 13 is a **Button** event. It displays a sprite and gives the viewer control of the sprite. This event is a multi-line event. Because there are a number of items that will occur during this event, it is carried over into extra lines. (Lines 13 through 17 are all connected to this Button event.)

---

*NOTE: You will need all fields for this Button event.*

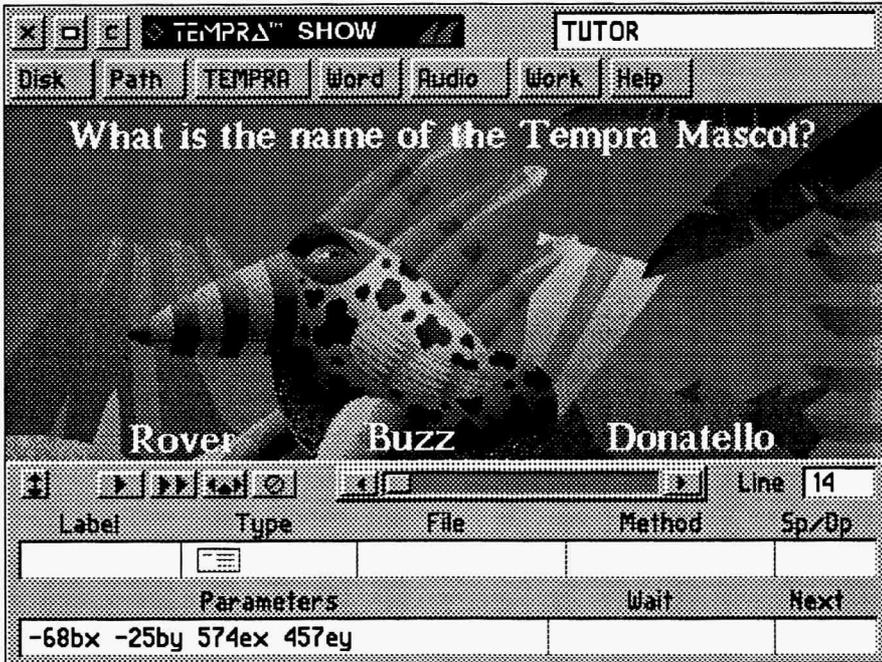
---

### **Recreating Line 13**

1. Select the box under **Label**. Type **quest** as the Label and press [Enter].
2. The **Type** menu pops up. Select **Button**.
3. The **File** menu pops up. Select **HAND.PCC** from the list.
4. The **Method** menu pops up. For **How Many Areas?**, type **3** and press [Enter].
5. The **SP/DP** menu pops up. Set the speed at **99** and **Erase**. Select **OK**.
6. The **Parameters** menu pops up. For **Insert Lines For Buttons**, select **Insert Lines**. The sprite appears with a message to position it at its starting point. Move the sprite to the **tip of Buzz's beak** and press **Select**.
7. The **Next** menu pops up. For **Next Label for <esc> on Button?**, type **quest**, press **Select**. (This tells TEMPRA SHOW to go back to the question if the viewer presses the [Esc] key.)
8. Another message appears in the upper left corner of the screen. This is for **Line 14**, so turn the page and continue.

# TEMPRA SHOW: Line 14, Button Event

*Objective: Continue the Button event started in Line 13.*



Line 14 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Text	Q1	Time30B	P 001	0x 471y 153c 73w 4h nrm		
	Sprite	HAND	Simple	15 E	-55bx 288by 142ex 288ey	2 0/18	
	Title		Time30B	P 001	Rover @30x 93y 153c		
	Title		Time30B	P 001	Buzz @200x 93y 153c		
	Title		Time30B	P 001	Donatello @375x 93y 153c		
quest	Button	HAND	3 Areas	99 T	142bx 288by -68bx -25by 574by 457ey		quest

Line: 14 Sequence: TUTOR F1-Help F5-Prev Line

## Line 14

Line 14 **sets the area** in which you can move the hand cursor. You can set a restrictive active area to prevent the hand from flying off the screen. Line 14 is a continuation of Line 13 and is marked by a special symbol in the Type field on the graphics menu.

---

*NOTE: You will not need the **Label, File, Method, SP/DP, Wait, and Next** fields for this part of the Button event.*

---

## Recreating Line 14

1. The **Parameters** menu was already up while you were in Line 13. This contained a message to position the cursor at the lower left corner of the active area. Move the sprite cursor to the **lower left corner of the screen** and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the **upper right portion** of the active area. Move the cursor to the **upper right corner of the screen**, and press **Select**.
3. Another message appears (the same as in the first message above). This is actually the beginning of **Line 15**. Turn the page and continue with the Button event.



## Line 15

Line 15 sets the active area that will branch to the “wrong” answer event. Line 15 is a continuation of Line 13 and is marked by a special symbol in the Type field on the graphics menu.

---

*NOTE: You will not need the Label, File, Method, SP/DP, and Wait fields for this part of the Button event.*

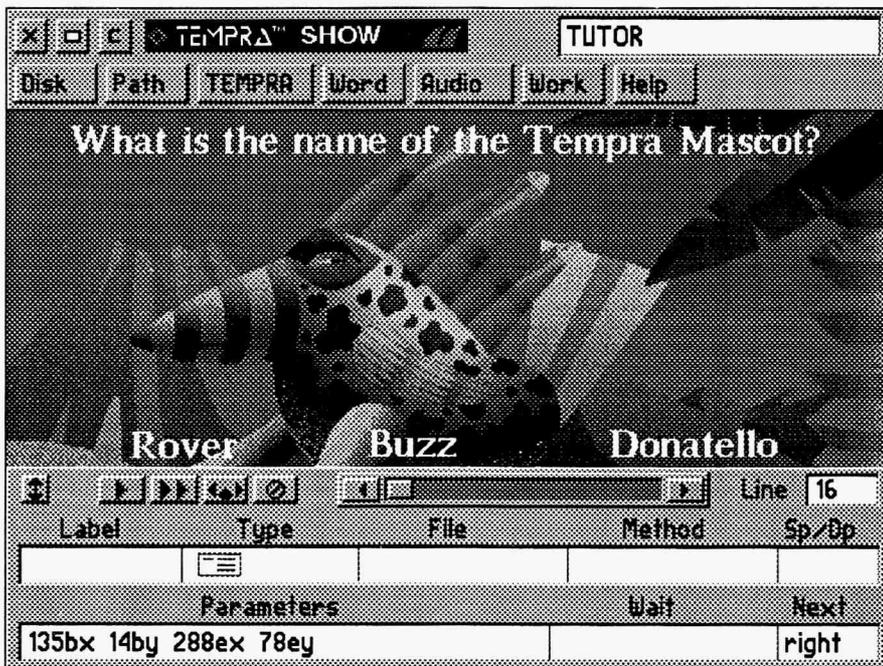
---

### Recreating Line 15

1. The **Parameters** menu was already up while you were in Line 14. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where the word “Rover” (or whatever word you used) is positioned. Move the cursor to the **lower left corner** of the word Rover and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the **upper right corner of the word Rover**. Press **Select**.
3. The **Next** menu pops up. For **Area 1**, type in **wrong** as the label. This causes a branch to the **wrong** label every time “Rover” is selected. At the **wrong** label during the sequence, a musical tune plays and text appears to indicate the wrong answer has been selected.
4. Another message appears (to determine another button area). This is actually the beginning of **Line 16**. Turn the page and continue.

# TEMPRA SHOW: Line 16, Button Event

*Objective: Continue the Button event started on Line 13.*



Line 16 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Title		Time30B	P 001	Rover @30x 92y 153c		
	Title		Time30B	P 001	Buzz @200x 93y 153c		
	Title		Time30B	P 001	Donatello @200x 93y 153c		
quest	Button	HAND	3 Areas	99 T	142bx 288by -68bx -25by 574by 457ey -34bx -25by 119ex 81ey -135bx 14by 288ex 78ey		wrong right

Line: 16 Sequence: TUTOR F1-Help F5-Prev Line

## Line 16

Line 16 sets the active area for the viewer to select that will branch to the “right” answer event. Line 17 is a continuation of Line 13 and is marked by a special symbol in the Type field on the graphics menu.

---

*NOTE: You will not need the Label, File, Method, SP/DP, and Wait fields for this part of the Button event.*

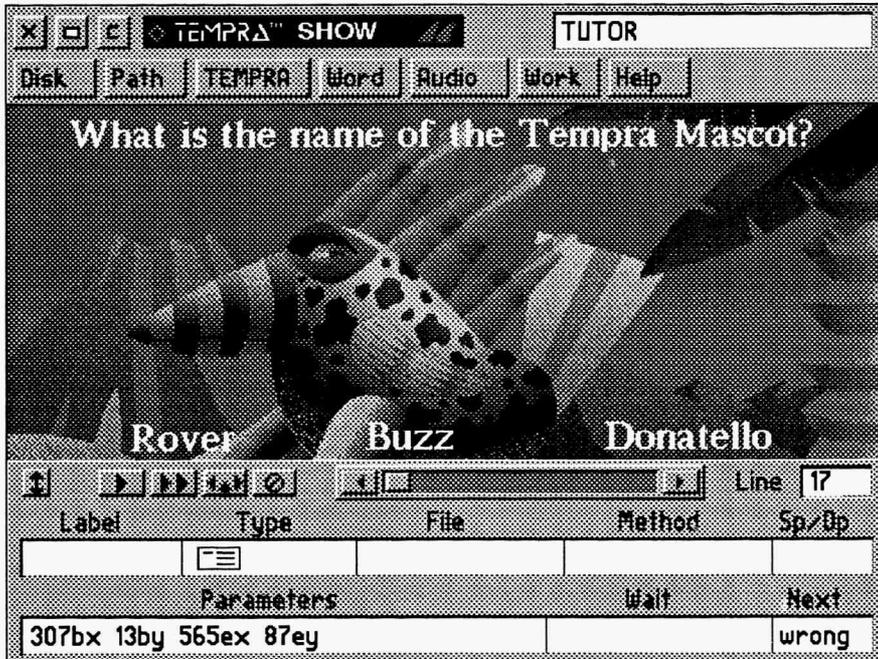
---

### Recreating Line 16

1. The **Parameters** menu was already up while you were in Line 15. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where the word “Buzz” is positioned. Move the cursor to the **lower left corner of the word Buzz** and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right of the active area. Move the cursor to the **upper right corner of the word Buzz** and press **Select**.
3. The **Next** menu pops up. For **Area2**, type in **right** as the label. This causes a branch to the **right** label when “Buzz” is selected. At the **right** label during the sequence, a musical tune plays and text appears indicating that the right answer has been selected.
4. Another message appears (to determine the third pointing area). This is actually the beginning of **Line 17**. Turn the page and continue.

# TEMPRA SHOW: Line 17, Button Event

*Objective: Continue the Button event started on Line 13.*



Line 13 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
	Title		Time30B	P 001	Rover @200x 93y 153c		
	Title		Time30B	P 001	Donatello @375x 93y 153c		
quest	Button	HAND	3 Areas	99 T	142bx 288by -68bx -25by 574by 457ey -34bx -25by 119ex 81ey -135bx 14by 288ex 78ey 307bx 13by 565ex 87ey		wrong right wrong

Line: 17 Sequence: TUTOR F1-Help F5-Prev Line

## Line 17

Line 17 sets the active area for the viewer to select that will branch to the “wrong” answer event. Line 17 is a continuation of Line 13 and is marked by a special symbol in the Type field on the graphics menu.

---

*NOTE: You will not need the Label, File, Method, SP/DP, and Wait fields for this part of the Button event.*

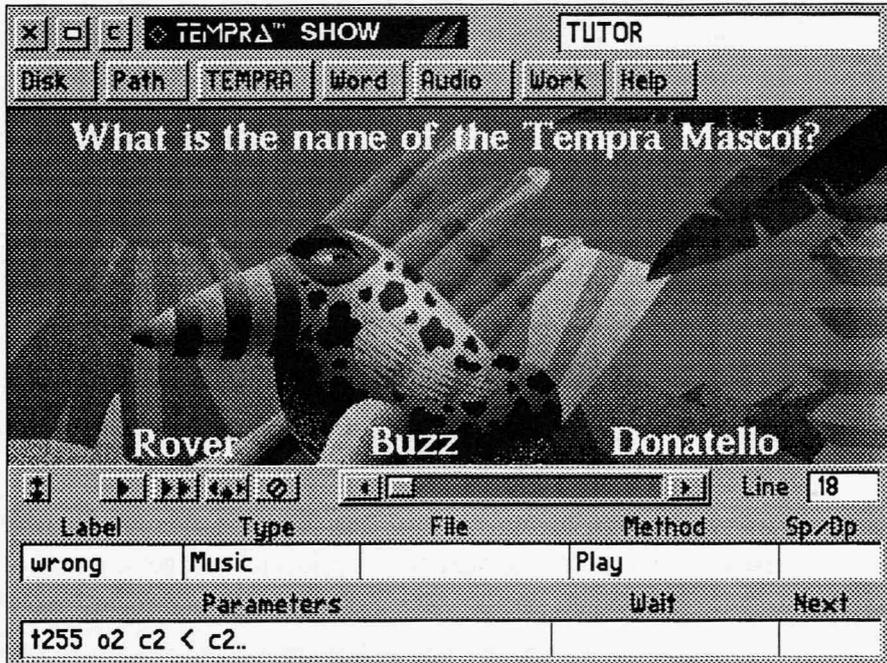
---

### Recreating Line 17

1. The **Parameters** menu was already up when you started Line 16. This contained a message to position the cursor at the lower left corner of the active area. The active area in this line is the location where the word “Donatello” (or whatever word you used) is positioned. Move the cursor to the **lower left corner of the word Donatello** and press **Select**.
2. Another message appears in the upper left corner of the screen asking you to position the cursor at the upper right portion of the active area. Move the cursor to the **upper right corner of the word Donatello**. Press **Select**.
3. The **Next** menu pops up. For **Area 3**, type in **wrong** as the label. The names in the Label and Next fields must match exactly, including upper and lowercase letters. In the tutorial sequence, they are all spelled with lowercase characters.
4. Press **Cancel** to return to the menus. You have completed the Button event for branching! The **Line Counter** automatically advances to Line 18. (If you happen to go past Line 18 when you cancel, rewind the Line Counter to Line 18 and pick up on the next page.)

# TEMPRA SHOW: Line 18, Music Event

*Objective: Create a short piece of music.*



Line 18 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
quest	Title	HAND	Time30B	P 001	Donatello @375x 93y 153c		
	Button		3 Areas	99 T	142bx 288by -68bx -25by 574by 457ey -34bx -25by 119ex 81ey -135bx 14by 288ex 78ey		wrong right wrong
wrong	Music		Play		307bx 13by 565ex 87ey t255 o2 c2 < c2..		

Line: 18 Sequence: TUTOR F1-Help F5-Prev Line

---

## Line 18

Line 18 is a **Music** event that begins the events that occur when the viewer selects the “wrong” answer.

---

*NOTE: You will not need to use the Files and SP/DP fields for this Music event.*

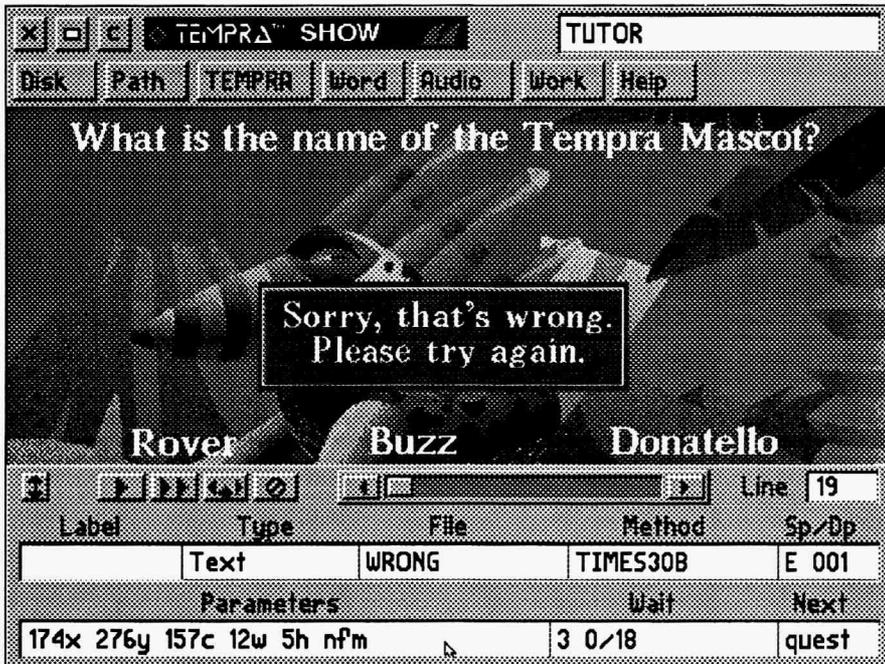
---

## Recreating Line 22

1. Select the box under **Label**. You need to assign a label to match the Next created in Lines 15 and 17. Lines 15 and 17 branch to Lines 18 and 19. Type **wrong** in the menu. (The spelling and case must match the Next fields in Lines 15 and 17.) Press **Select** to accept.
2. The **Type** menu pops up. Scroll down and select **Music**.
3. The **Method** menu pops up. Select **Play**, since you are going to create a few notes of music on the keyboard that will play during this branch.
4. The **Parameters** menu pops up. This is where you type the music notes to play through the computer’s speakers. The notes we created are t255 o2 c2 < c2... Type **t255 o2 c2 < c2..** in the menu and press **Select**. See Appendix D for information about creating music on the computer keyboard.
6. The **Wait** menu pops up. Select **Don’t Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. The “wrong” answer music plays. The **Line Counter** automatically advances to Line 19.

# TEMPRA SHOW: Line 19, Text Event

*Objective: Create erasing framed text.*



Line 19 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
quest	Button	HAND	3 Areas	99 T	142bx 288by -68bx -25by 574by 457ey -34bx -25by 119ex 81ey -135bx 14by 288ex 78ey 307bx 13by 565ex 87ey		wrong right wrong
wrong	Music		Play		+255 o2 c2 < c2..		wrong
	Text	WRONG	Time30B	E 001	174x 276y 157c 12w 5h nfm	3 0/18	quest

Line: 19 Sequence: TUTOR F1-Help F5-Prev Line

## Line 19

Line 19 contains the **framed text** “Sorry, that’s wrong. Please try again.” This text event is a little different from the earlier ones.

---

***NOTE:** You will not need to use the **Label** field.*

---

### Recreating Line 19

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **WRONG.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you want to use or scroll down and select **Times30**.
4. The **SP/DP** menu pops up. Select **Erase After Displaying**.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select the first line of the words “Sorry, that’s wrong...” For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image?**, select **Yes**. Select a color from the image using the cross-shaped cursor.
6. Framed text appears in the top left corner. **Move the frame around the text you want to display** (in this case, “Sorry, that’s wrong. Please try again.”) and press **Select**. This generates a frame for the text. The frame appears on the screen for positioning. Move the frame to the **center** of the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait For Timeout**. In the Seconds box, type **3**. Press **Select**. Press **Select** again to accept 18th Seconds default.
8. The **Next** menu automatically pops up. Type **quest**. (This branches the sequence back to Line 13, which is labeled **quest**, so you can select the correct answer.)
9. Framed text appears on the screen. The **Line Counter** automatically advances to Line 20.

# TEMPRA SHOW: Line 20, Music Event

*Objective: Create a short piece of music.*

Label	Type	File	Method	Sp/Dp	Parameters	Wait	Next
right	Music		Play		t255 o4 c8 d8 e8 f8 g		

Line 20 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
					-68bx -25by 574by 457ey -34bx -25by 119ex 81ey -135bx 14by 288ex 78ey 307bx 13by 565ex 87ey		wrong right wrong
wrong	Music		Play		t255 o2 c2 < c2..		
	Text	WRONG	Time30B	E 001	174x 276y 157c 12w 5h nfm	3 0/18	quest
right	Music		Play		t255 o4 c8 d8 e8 f8 g		

Line: 20 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 20***

Line 20 is a **Music** event that begins the events that occur when the viewer selects the “right” answer.

---

*NOTE: You will not need to use the Files and SP/DP fields for this Music event.*

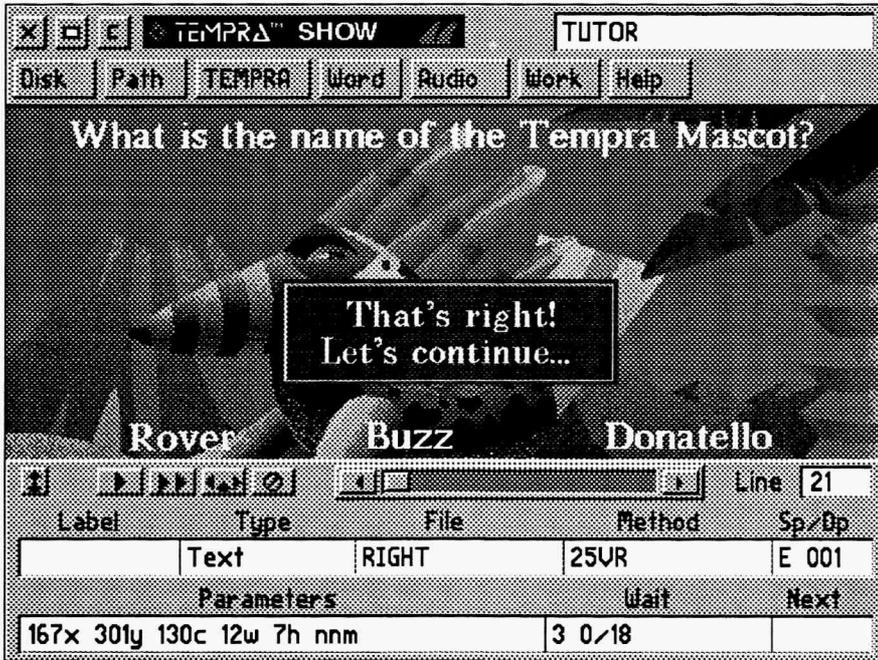
---

### **Recreating Line 20**

1. Select the box under **Label**. Type **right** in the menu. Press **Select** to accept.
2. The **Type** menu pops up. Scroll down and select **Music**.
3. The **Method** menu pops up. Select **Play**.
4. The **Parameters** menu pops up. Type **t255 o4 c8 d8 e8 f8 g** in the menu and press **Select** (these are musical notes to play during the “right answer” event).
6. The **Wait** menu pops up. Select **Don’t Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. The “right” answer music plays. The **Line Counter** automatically advances to Line 21.

# TEMPRA SHOW: Line 21, Text Event

*Objective: Create erasing framed text.*



Line 21 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
					-34bx -25by 119ex 81ey -135bx 14by 288ex 78ey 307bx 13by 565ex 87ey		wrong right wrong
wrong		WRONG	Play		t255 o2 c2 < c2..		
right	Music		Time30B	E 001	174x 276y 157c 12w 5h nfm	3 0/18	quest
	Text	RIGHT	Time30B	E 001	167x 301y 130c 12w 7h nfm		

Line: 21 Sequence: TUTOR F1-Help F5-Prev Line

## Line 21

Line 21 contains the **framed text** “That’s right! Let’s continue...” You can treat this Text event the same as in Line 19.

---

*NOTE: You will not need to use the **Label field**.*

---

### Recreating Line 21

1. Select the box under **Type**. From the pop-up menu, select **Text**.
2. The **File** menu pops up. Select **RIGHT.TXT** from the list.
3. The **Method** menu pops up. Select **KST Fonts**. Select a typeface you want to use or scroll down and select **Times30**.
4. The **SP/DP** menu pops up. Select **Erase After Displaying**. This prompts the text to disappear after it is displayed, rather than waiting for the next event to wipe it off the screen.
5. **Parameters** pops up. For **Choose Starting Line For Text**, select the **first line** in “That’s right! Let’s continue..” For **Allow User to Operate?**, select **No**. For **Draw Frame**, select **Yes**. For **Justify**, select **Middle**. For **Choose Color From Image**, select **Yes**. Using the cross-shaped cursor, select a color from the image.
6. Framed text appears in the top left corner. **Move the frame around the text you want to display** (in this case, “That’s right! Let’s continue..”) and press **Select**. Move the text to the **center** of the screen and press **Select**.
7. The **Wait** menu pops up. Select **Wait For Timeout**. In the Seconds box, type **3**. Press **Select**. **Select** again to accept 18th Seconds default.
8. The **Next** menu pops up. Press **Select** to accept the default.
9. Text appears in the box. The **Line Counter** advances to Line 22.

# TEMPRA SHOW: Line 22, BColor Event

*Objective: Wipe a black background onto the screen.*



Line 22 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
wrong			Play		-135bx 14by 288ex 78ey 307bx 13by 565ex 87ey t255 o2 c2 < c2..		right wrong
right	Music	WRONG	Time30B	E 001	174x 276y 157c 12w 5h nfm	3 0/18	quest
	Text	RIGHT	Time30B	E 001	167x 301y 130c 12w 7h nfm		
end	BColor		Box_Out	99	Solid 0c		

Line: 22 Sequence: TUTOR F1-Help F5-Prev Line

## **Line 22**

Line 22 is a **BColor** event that changes the background color to black as a transition to the final image screen.

---

***NOTE:** You will not need the **File** field for this **BColor** event.*

---

### **Recreating Line 22**

1. Select the box under **Label**. Type **end**.
2. The **Type** menu pops up. Select **BColor**.
3. The **Method** menu pops up. From the **Wipe** menu, select a wipe you want to try or select **Box\_Out** (this makes a wipe that spirals from the center of the screen to the outside).
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK** to accept the speed.
5. The **Parameters** menu pops up. On the **Choose Color From Image?** menu, select **No**. The color palette appears. Select the black square (**0**) and then **OK**.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. The screen color changes to black. The **Line Counter** advances to Line 23.

# TEMPRA SHOW: Line 23, Video Event

*Objective: Run an animated .FLI file.*



Line 23 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
					-135bx 14by 288ex 78ey 307bx 13by 565ex 87ey		right wrong
wrong		WRONG	Play		t255 o2 c2 < c2..		
			Time30B	E 001	174x 276y 157c 12w 5h nfm	3 0/18	quest
right	Music		Play		t255 o4 c8 d8 e8 f8 g		
	Text	RIGHT	Time30B	E 001	167x 301y 130c 12w 7h nfm		
		TOUCAN2	AaPlay	95	83bx 364by		

Line: 23 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 23***

Line 23 is a **Video** event that plays an animated (.FLI) file of Buzz.

---

***NOTE:** You will not need the **Label** field for this Video event.*

---

### **Recreating Line 23**

1. The **Type** menu pops up. Select **Video**.
2. The **Method** menu pops up. Select **Aaplay** at the bottom of the menu.
3. The **Files** menu pops up. Select **TOUCAN2.FLI** from the menu.
4. The **SP/DP** menu pops up. Set the speed at **95**. Select **OK** to accept the speed.
5. The **Parameters** menu pops up. A ghost rectangle appears for you to position the viewing window on the screen. Move the rectangle to the **center** of the screen and press **Select**. The **Number of Repeats** menu pops up. Press **Select** to accept the default of 1.
6. The **Wait** menu pops up. Select **Don't Wait**.
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. The animated file of Buzz spinning plays. The **Line Counter** advances to Line 24.

# TEMPRA SHOW: Line 24, Screen Event

*Objective: Wipe a .PCX image on the screen.*



Line 24 completed.

TEMPRA SHOW							
Label	Type	File	Method	Sp/Dp	Parameters	Wait For	Next
wrong		WRONG	Play	E 001	t255 o2 c2 < c2..	3 0/18	quest
right	Music		Play	E 001	t255 o4 c8 d8 e8 f8 g		
	Text	RIGHT	Time30B	E 001	167x 301y 130c 12w 7h nfm		
end	BColor		Box_Out	99	Solid 0c		
	Video	TOUCAN2	Aaplay	95	83bx 364by		
	Screen	SLIDE20	Dissolve	99		any key	

Line: 24 Sequence: TUTOR F1-Help F5-Prev Line

## ***Line 24***

Line 24 is a **Screen** event with a .PCX image of the TEMPRA product box. This is similar to the other image files you loaded earlier.

---

**NOTE:** *You will not need the **Label** and **Parameters** fields for this Screen event.*

---

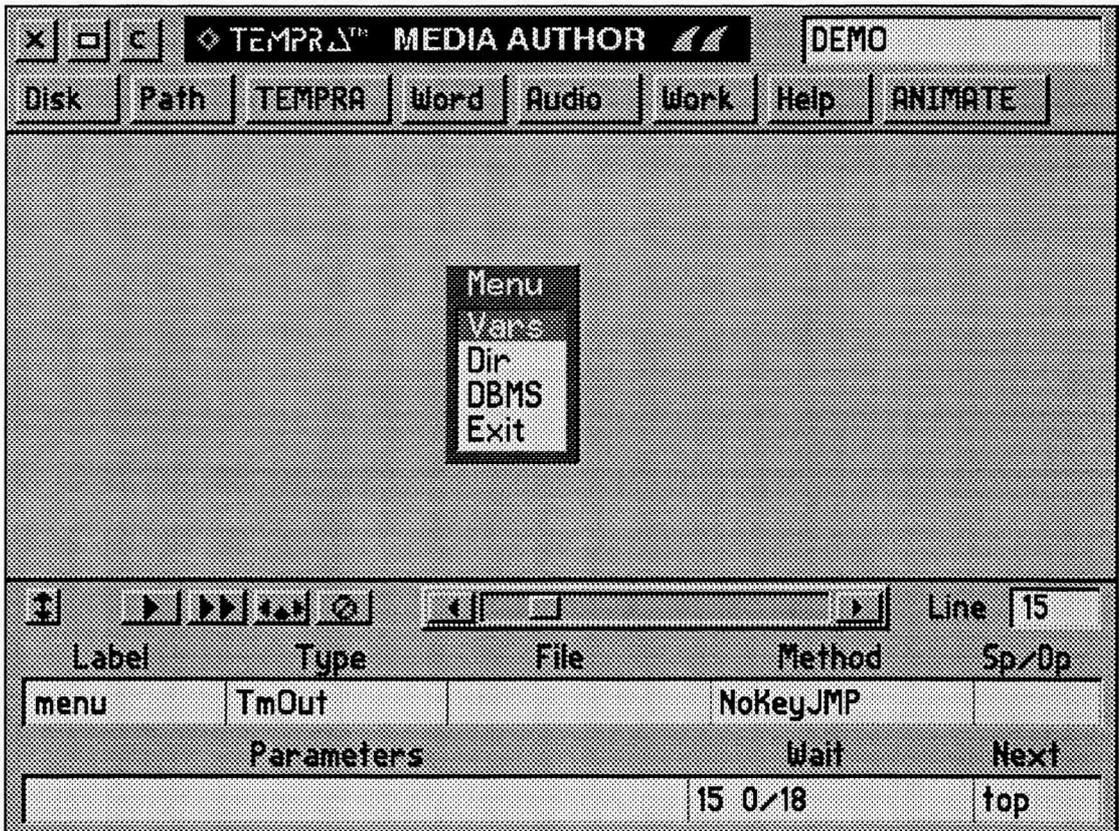
### **Recreating Line 24**

1. Select the box under **Type**. From the pop-up menu, select **Screen**.
2. The **File** menu pops up. Select **SLIDE20.PCX** from the list.
3. The **Method** menu pops up. From the **Wipe** menu, select a wipe you'd like to try or select **Dissolve**.
4. The **SP/DP** menu pops up. Set the speed at **99**. Select **OK** to accept the speed.
5. The **Parameters** menu pops up. For the **Show Screen** menu, select **As Is**.
6. The **Wait** menu pops up. Select **Wait For Any Key**. (This prompts the sequence to start over again when the viewer presses a key on the keyboard or presses **Select** on the mouse.)
7. The **Next** menu automatically pops up. Press **Select** to accept the default.
8. An image of the product box dissolves onto the screen. The **Line Counter** automatically advances to Line 25.
9. To watch the sequence you just created, rewind to **Line 1** and press **[F6]**.

### **Congratulations!**

**You have completed the tutorial. Have fun with TEMPRA SHOW!**

*The staff of Mathematica, Inc.  
Committed to Imaging Excellence*



## Appendix J

# ***TEMPRA Media Author Tutorial***

This tutorial will give you some idea of the power of TEMPRΔ Media Author. It includes many events you used in the TEMPRΔ Show tutorial, such as **BColor**, **Global**, **Menu**, **Screen**, **Text**, **Title**, and **Video**. But it also includes events found only in TEMPRΔ Media Author, such as the **DBMS**, **Dir**, **IfThen**, **TmOut**, and **Var** events.

## ***Tutorial Objectives***

The primary objective is to simply provide an opportunity for you to actually use the program to create a useful sequence. Hopefully, you can then draw on the experience you gain from the tutorial as you develop your own applications.

### ***A Menu of Choices***

The other objectives deal specifically with using the events of TEMPRA Media Author interactively with the events already found in TEMPRA Show. We start by building a **Menu** from which users can make a selection. Users choose from **Var**, **Dir**, **DBMS**, and **Exit**. The sequence then branches to a specific area that demonstrates the particular selection. Each series of events terminate by returning to the opening **Menu**.

### **Menu Item 1 — Var**

The first **Menu** selection demonstrates the use of variables. In creating this portion of the sequence, we show you how to use the **Var** event to declare and modify a string variable that holds a name, then combine the use of the variable with a **Title** event to ask users to enter their name. A numeric variable keeps track of how many times they enter a name. The sequence then uses the **IfThen** event to test these variables. The string variable is tested for a specific name, while the numeric variable is tested against a predetermined number. When either of the **IfThen** test become true, the sequence returns to the opening **Menu**.



Sharp Boy says:  
I use numeric variables to help keep track of the royalties I receive from appearing in this manual.

A practical application for this may be in the development of computer-based training. Students are asked to enter their names, the sequence compares the name they enter with an existing list of students and then branches to the portion of the sequence that applies specifically to each student.

### **Menu Item 2 — Dir**

The second **Menu** selection demonstrates the use of the **Dir** event used in conjunction with **Screen**, **Video**, **Var**, and **IfThen** events. This example displays a list of files that are located in the TEMPRA **Screen** directory. The sequence locates the **Screen** directory by using the system variable for



Shapr Boy says:  
I can't always remember how each  
image in my screen directory looks.  
By using the **DIR** event, I can easily  
check any image I have on file.

**Screen Directory.** The files are listed in a **Menu** titled “**Images,**” and are shown in alphabetical order. When the users select one of the files from the **Menu**, a series of **IfThen** events determine the type of file selected, and the sequence branches to a **Screen** or **Video** event to display the selected file on the screen.

When creating an application with TEMPRA Media Author, the **Dir** event could be used to quickly view images that are to be included in the sequence.

Instead of displaying image files from the **Screen** directory, the sequence could have been directed to display a list of program files in some other directory. Then a **System** event could be used to allow the users to execute a particular program. In this manner, TEMPRA Media Author acts as a DOS shell program.

### Menu Item 3 — DBMS

The third **Menu** selection demonstrates the database capabilities of TEMPRA Media Author. It combines the **DBMS** event with the **IfThen**, **Var**, **Screen**, and **Video** events. A database file named “**IMAGES**” is opened, and the fields of each record are read into the variable table. TEMPRA Media Author then reads the variables to determine the type of file, then branches to either a **Screen** or **Video** event to display the file on the screen. A **Var** event is also used to display the name of the file being shown on the screen, along with a comment about the file.



Shap Boy won't tell us what kind of  
images are in his database. He  
says the pictures are private.

As each record is read, and each file displayed, the record counter changes and the sequence loops to read the next record. An **IfThen** event compares the record number currently being read, and when all files listed in the database have been displayed, the sequence branches back to the opening **Menu**.

For someone who already uses a database of image files, TEMPRA Media Author interfaces with this database to display the images listed in the existing database.

The tutorial combines all of these events together with text and titles to provide a colorful presentation. **TmOut** events are used to keep the sequence from stalling.



Sharp Boy says:  
The workflow can save a lot of time for the experienced user. The novice will find it easier to follow the detailed instructions.

These instructions start off being very detailed, and as events are repeated, you will be told only what is necessary to fill in each **Menu** item. If you have trouble with the specifics of how to do a particular task, refer to a previous event line. Use the default for **Wait** and **Next** unless otherwise specified. All events begin in the **Type** field unless otherwise specified.

### **A Shortcut**

If you're already experienced at creating sequences and do not necessarily need step-by-step instructions, you may want to simply refer to the workflows presented throughout the tutorial. The workflow shows the contents of each field for each event line in the sequence.

## **Tutorial Conventions**

---

To help you recognize the prompts from TEMPRA Media Author, and to know what you are expected to enter, the following conventions are used throughout this tutorial.

When TEMPRA Media Author asks you a question or to make a **Menu** selection, the text will be shown in **bold** type as it appears on the Screen. If you see in the text **Input Title:**, this means TEMPRA Media Author is asking you to enter text that will be used for a **Title** event. **Field** and **Event** names are also shown in **bold type**.

When you see bold text in quotation marks, this tells you what you should type in response to an expected entry or shows you something you have already entered. If you see in the text **enter “\_file”** this means you should type the underscore character followed by the word “file”. If you see in the text **enter “\$path”** this means you should type the dollar sign character followed by the word “path”. Pay particular attention to what is between quotation marks and enter it exactly as it is shown in the text.

**Line #** - Step by step instructions for each line are preceded by the line number in bold text. This is where we tell you “enter this” or “select that.”

*Text that is used to explain an event is shown in italics.*

When you see [Enter], this refers to the [Enter] key on your keyboard or the [Select] (left) button on your mouse.

### ***Before You Exit***

Because the tutorial is quite extensive, you may find it necessary to stop and start several times. Make sure you save the sequence each time before you exit Tempra Media Author. When you come back to the program, just load the sequence from the **Disk** menu. After you load the sequence, use the [F6] key to run the lines already created. This will make sure all parameters are in place, and you can then use the event Line Counter to advance to where you left off in the sequence.

## ***Getting Started***

Move your cursor to the **Disk Button** on the left side of the **Menu Bar** and press [Enter]. Select **New Sequence** and enter a name for this sequence, such as “**DEMO.**” Make sure you see a “1” in the Line Counter on the event controller. This tells you that you are starting on line one.

### **Lines 1 - 5**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Bcolor		Quick	1	Solid 0c		
	Global		ResetAll				
	Global	COUR20	Font				
	Video		SetMode		SGA_2E 640x480 256		
	Video	TUTOR1	LoadPal				

*The first few event lines make sure you start with “known operating conditions.” Start by making the Screen black.*

### **Line 1**

Move your cursor to the **Type** field and press [Enter], select **BColor**, and select **Quick**. The speed doesn’t matter when using **Quick**, so just select

**OK** on the **Sp/Dp** menu. When asked **Choose a Color From Your Image?**, select **No**. From the color palette select color **0**, which is black. Select **Don't Wait**, and press **[Enter]** in the **Next** field.

*You will notice as each line is completed, TEMPRA Media Author executes that line and advances the Line Counter. However, some events do not actually change anything on the screen. For example, if your screen is already black, you will not see it change in the above event. Also, when you declare or modify a variable, there is no effect on the screen.*

### **Line 2**

Position the cursor in the **Type** field and press **[Enter]**. Scroll down and select **Global**, select **Reset All**, and select defaults for the rest of this line.

### **Line 3**

With the cursor again in the **Type** field, press **[Enter]**, select **Global**, select **Font**, select **KST Font**, and select **COUR20**. Accept defaults for the remainder of this line.

*Any time the default font is used in the sequence, such as for the **Var Input** event, that font is set to **COUR20** by the above event.*

### **Line 4**

Select **Video**, select **SetMode**, and select **SGA\_2E 640 x 480 256**.

*Even though your monitor may support a higher resolution and color palette, choose this one so that the screens in the sequence are consistent with the expected results.*

### **Line 5**

Select **Video**, select **LoadPal**, and select **TUTOR1.PCX** as the file.

*You should now see the menu screen change color to reflect the new color palette that was just loaded.*

You can use a different font, mode, and color palette than the ones shown above, but your results maybe somewhat different. For now, stay with



Sharp Boy says:  
To ensure predictable results,  
always declare variables at the  
beginning of the sequence.

selections we suggest, and after you've completed the sequence and see how everything works, you can go back and make changes as you like.

*In the next series of events, you will declare all of the variables that will be used in the sequence. For best results, always declare variables at the beginning of a sequence, and modify them later as needed.*

### Lines 6 - 9

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
devar	Var		Declare		\$name - " "		
	Var		Declare		#count - 0		
	Var		Declare		\$_file - " "		
	Var		Declare		\$path - " "		

#### Line 6

With your cursor in the **Label** field, press [Enter] and enter a name for the group of vars, such as “devar”. From the **Type** field, select **Var**, and in the **Method** field, select **Declare**. Since this variable will be used for entering a person’s name, select **String Var**. For **Input a Var Name : \$** enter “name”. For **Initial value**, simply press the spacebar one time and then press [Enter]. This sets the initial value of this string variable to be blank. Accept the default for the **Wait** and **Next** fields. Your **Parameters** field should now read: `$name = " "`.

*Notice that TEMPRA Media Author put quotes around a blank space. This indicates that the initial value of the string var is blank.*

*The next variable we want to declare is for counting how many times a particular event occurs.*

### Line 7

In the **Type** field, select **Var**, then select **Declare**, select **Numeric Var**, enter “**count**” as the name, and set the initial value to **0**. Press **[Enter]** to accept the defaults for the remainder of this event.

*The next variable will be used in conjunction with a **Dir** event and holds a filename to be used in a **Screen** or **Video** event.*

### Line 8

In the **Type** field, select **Var**, then select **Declare**, select **String Var**. For **Input a Var Name**: \$ enter “**\_file**”. For **Initial Value**, press the spacebar one time and then press **[Enter]**. The **Parameters** field should read: \$**\_file** = “ ”.

*We need one more variable to hold the DOS pathname we want to use in selecting our files with the **Dir** event.*

### Line 9

In the **Type** field Select **Var**, then select **Declare**, select **String Var**. Enter “**path**” for the **Var Name**. For **Initial Value**, press the spacebar one time and then press **[Enter]**. Your parameters field should read: \$**path** = “ ”.

## Play A Video

*The next series of events combine **Text** events with a **Video** and **TmOut** event. This is actually the first thing that appears on the screen when the sequence is run. While **Buzz** spins around on the screen, we will use a **Text** event to instruct users to press the **[Enter]** key to view the **Menu**. Actually, any key pressed causes the sequence to branch to the opening **Menu**. This demonstrates a practical use for the **TmOut** event.*



Sharp Boy never presses the **[Enter]** key when he runs this sequence. He likes to keep **Buzz** spinning.

## Lines 10 - 14

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
top	Tmout		KeyJMP				menu
play	BColor		Dissolve	95	Solid 63c		
	Text	MA	TIMES25	T 001	142x 428y 226c 14w 3h nfm		
	Text	View		T 001	138x 170y 226c 25w 3h nfm		
a	Video	TOUCAN2	AaPlay	90 T	154bx 365by 1R		a

### Line 10

With your cursor in the **Label** field, press [Enter] and type “top”. Select **TmOut**, and select **KeyJMP** for the **Method**. Enter “menu” in the **Next** field.

*This event tells TEMPRA Media Author to branch to the label “menu” when any key is pressed.*

### Line 11

In the **Label** field, press [Enter] and type “play” to give this series of events a label. *This label is necessary since we branch to it later in the sequence.* In the **Type** field, select **BColor**, select **Dissolve**, and set the **Speed** to **90**. When asked if you want to **Choose a Color From Your Image?**, select **No**. This brings up the color palette from which you should choose color number **63**.

Assuming you created the **Video LoadPal** event in Line 5 using **TUTOR1** as the file, color 63 should give us the perfect color for our demonstration.

### Line 12

In the **Type** field, select **Text**. For the filename select **MA.TXT**. Select the **KST Font TIMES25**, and for **Display** select **Temporary**. Now use your mouse to point to the first line of text and press [Enter]. Answer **No** to **Allow User to Operate?** and answer **Yes** to **Draw Frame?**, then select **Middle** for **Justify**.



Sharp Boy says:

Use the mouse to position text close to where you want it to go. Then, by using your cursor keys, you are able to move in much smaller increments to fine-tune the desired placement.

When asked to **Choose a Color From Your Image?**, select **No**. Choose color number **226** from the color palette. This color may look green right now, but turns red when the sequence is run and the **Video** event plays.

Now use your mouse to position the frame around the **Text** and press **[Enter]**. You should then see a large gray box and message in the upper left hand corner of the screen directing you to position the box. If you use **[Ctrl]-X** to display the x,y coordinates, you can position the box at coordinates 142x by 428y. This locates the box in the center of the screen, about one inch from the top. When you have the box where you want it, press **[Enter]**.

*When you have completed the above line, the Line Counter advances to Line 13. Move the Line Counter back to Line 12 and take a look at the Parameters field. If the x,y coordinates are not close to the ones we suggest, there is a simple way to change them. Position the pointer in the Parameters field and press [F3]. The field changes color and you'll see a flashing cursor. You can now use your cursor keys to move the cursor to the coordinates listed in the field, then simply enter the new coordinates from the keyboard and then press [Enter].*

*To see the new position on the screen, select the **Play Line** button on the event controller. (This is the single arrow.) TEMPRA Media Author plays the line, showing the new location of the text.*

Before going on to the next event, advance the Line Counter to 13.

### **Line 13**

From the **Type** field, select **Text**, select **VIEW.TXT** for the filename. Select **Default Font** for this text, and for **Display**, select **Temporary**. Now use your mouse to point to the first line of text and press **[Enter]**. Answer **No** to **Allow User to Operate?**, answer **No** to **Draw Frame?**, and select **Middle** for **Justify**.

When asked to **Choose a Color From Your Image?**, select **No**. Again you see the color palette. Choose the same color **226** for this text.



Sharp Boy says:  
If you know the exact x,y coordinates you want to use for positioning on the screen, entering the coordinates directly into the **Parameters** field can be much faster than moving boxes around on the screen.

Now use your mouse to position the frame around the text and press **[Enter]**. You should see the text near the bottom of the screen and a message in the upper left-hand corner of the screen directing you to position the box. Use **[Ctrl]-X** to display the x,y coordinates and position the box at coordinates 138x by 170y. This locates the text near the center of the screen, about two inches from the bottom. When you have the text where you want it, press **[Enter]**.

*The next event plays a video file in the center of the screen, between the two bodies of text created in Lines 12 and 13. We created a **TmOut** event in Line 10 using the **KeyJMP Method**, because we want the **Video** to continue playing until someone presses the **[Enter]** key. Therefore, we will build a simple loop.*

#### Line 14

Press **[Enter]** in the **Label** field to give this event a name, such as “a”. In the **Type** field, scroll down and select **Video**, then select **AaPlay**, and select **TOUCAN2.FLI** for the filename. Set the speed to **95** and select **Permanent**. Next you see a message in the upper left hand corner of the screen directing you to position the cursor at the upper left corner of the screen. As you start to move your cursor, you should see a large rectangle moving across the screen. This is the box in which the video will be seen. Position it between the two lines of text already on the screen, at coordinates 154bx by 365by.

For the **Number of Repeats?** enter “1”, and in the **Next** field type “a” (this matches the name you typed in the **Label** field). By doing this, the event loops back to itself and the video continues to play until a key is pressed on the keyboard.

### **Good Housekeeping**

*Later in this sequence, a number of different **Video** and **Screen** events take place when the users make a selection from the **Menu**. Each series of events terminate by branching back to the **Menu**. To make sure that our colors and palette remain consistent each time a new selection is made from the **Menu**, we will enter a few “housekeeping” events.*

## Lines 15 - 20

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
menu	TmOut		NoKeyJMP			15 0/18	top
	BColor		Quick		Solid 0c		
	Video		SetMode		SGA_2E 640x480 256		
	Video	TUTOR1	LoadPal				
	Global		Colors		121 236B 98 43T 121 98H		
	BColor		Dissolve	99	Dolid 29c		

*In Line 10 we entered a **KeyJMP TmOut** event. The **TmOut** event branches to a label named “**menu**” in Line 15.*

### Line 15

Type “**menu**” in the **Label** field. So that the sequence doesn’t stall with the **Menu** on the screen, select another **TmOut** event from the **Type** menu, select **NoKeyJMP** as the **Method**, and enter “**15**” seconds, and enter “**top**” as the label in the **Next** field.

### Lines 16

Now we want make the screen black. Select a **BColor** event from the **Type** menu, scroll down and select **Quick** as the **Method**, and select color number **0**. Remember that speed does not matter when using the **Quick** wipe.

### Line 17

Now select a **Video** event, select **Set Mode**, and scroll down to select **SGA\_2E 640x480 256**.

### Line 18

From the **Type** field, select **Video**, select **LoadPal**, and select **TUTOR1** as the filename. You may recognize these two lines as being repeated from the top of the sequence. This simply returns us to a common operating environment each time we reenter the **Menu**.

---

*Because of the color palette being changed, the **Menu** we build may be hard to see unless we change the default colors of **TEMPRA Media Author**. Rather than switching back and forth between the example **Menu** and the color palette, we'll take a shortcut to changing the **Menu** colors for now. You can manually change them later to suit your individual taste.*

### **Line 19**

In the **Type** field, select **Global**, and then select **Colors**. When you see the **Menu** choices for **Border Foreground**, **Border Background** etc., press the [Esc] key. Move your cursor into the **Parameters** field and press [F3]. You'll notice the box changes colors, and you can now simply type in the numbers for the colors you want to use. Type "**121 236B 98 43T 121 98H**" and press [Enter].

*The above numbers "**121 236B**" represent the foreground and background colors respectively for the borders. The numbers "**98 43T**" represent the foreground and background colors for text. The numbers "**121 98H**" represent the foreground and background colors for highlighted text.*

Before you can enter the next event line, you must advance the Line Counter to Line 20 for the next event.

### **Line 20**

In the **Type** field, select **BColor**, select **Dissolve**, set the speed to **99** and select color **29** from the color palette.

## ***The Opening Menu***

*Now we are ready to build the **Menu**. The **Menu** provides users with four choices from which to select. When a selection is made from the **Menu**, **TEMPRA Media Author** branches to another part of the sequence to demonstrate the users' selection. When that series of events has executed, the sequence returns to the **Menu**.*

### Lines 21 - 26

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Menu		4 Items	P	252x 329y 5w 6h		
					Menu		top
					Vars		vars
					Dir		dir
					DBMS		dbms
					Exit		exit

*In creating the **Menu** event, the Line Counter automatically advances for Lines 21 - 26. Unless you watch the Line Counter, you may not be aware of this. When you create a **Menu** event, TEMPRA Media Author connects each of the lines needed to create the **Menu** event. Therefore, rather than listing the individual line numbers, just follow the instructions in order. If you make a mistake while creating the **Menu** event, you will want to start over at Line 21.*

### Lines 21 - 26

In the **Type** field, select **Menu**. The number of **Menu** items will be **4 items**, select **Permanent**, and select **Insert Lines**. For **Menu Title**, enter "**Menu**" and for **Next Label** for **<esc> on Menu?** enter "**top**". This tells TEMPRA Media Author where to branch to if the user presses the [Esc] key while the **Menu** is on the screen.

For **Menu** item one, enter "**Vars**" and enter "**vars**" for the **Menu** item one **Next Label**. The **Next Label** tells TEMPRA Media Author where to branch to if **Vars** is selected from the **Menu**.

For **Menu** item two, enter "**Dir**" and enter "**dir**" for the **Next Label**.

For **Menu** item three, enter "**DBMS**" and enter "**dbms**" for the **Next Label**.

For **Menu** item four, enter "**Exit**" and enter "**exit**" for the **Next Label**.

You will now see the **Menu** appear on the screen. Notice the **Menu** has a title at the top: **Menu**. Use your mouse or cursor keys to draw a frame around the **Menu** items, then position the **Menu** in the center of the screen.

*Now that the **Menu** itself is in place, TEMPRA Media Author needs to know what to do when **Menu** items are selected. When the **Menu** event was created in Lines 21 - 26, you should have entered “**Var**” for item one, and “**vars**” as the **Next Label**. The label that we enter for the following series of events needs to match exactly the **Next** label assigned in the **Menu** event.*

### **Menu Item 1 — Var**

*In creating the **Menu** event in Lines 21 - 26 above, we entered **Var** for the first **Menu** item. The **Next Label** that we entered for item one was “**vars**”. In Line 27 below we will enter “**vars**” in the **Label** field. This is where the sequence branches to when **Vars** is selected from the **Menu**.*

### **Lines 27 - 30**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
vars	BColor		Hrz_In	55	Solid 29c		
	Title		FUTURA20	T	Enter @55x 415y 191c		
	Title		FUTURA20	T	Your @125x 415y 191c		
	Title		FUTURA20	P	Name Here@189x 415y 191c		

### **Line 27**

Press [Enter] in the **Label** field, type “**vars**” and press [Enter]. For **Type**, select **BColor**, select **Hrz\_In**, set speed to “55” and from the color palette, select color number **29**. Select **Don’t Wait** and press [Enter] for **Next**.

*We want to display text on the screen that says “**Enter Your Name Here.**” You can do this with a **Text** event, which requires a corresponding text file, or you can use the **Title** event. The TEMPRA Media Author **Title** event*

*limits you to nine characters, so we will use three separate **Title** events. After you have completed this tutorial, experiment by creating a **Text** file to use with a **Text** event in place of the three individual **Title** events below.*

### **Line 28**

In the **Type** field, select **Title**, select **KST Font FUTURA20**, and make it **Temporary**. For **Input Title** type “**Enter**” and choose color number **191** from the color palette. Position it at x,y coordinates 55x by 415y.

*When positioning text on the screen at specific x,y coordinates, you may find it easier to use the mouse to move the block of text to the general location, and then use the cursor keys on the keyboard to position it at the precise x,y coordinates.*

### **Line 29**

In the **Type** field, select **Title**, select **KST Font FUTURA20**, and make it **Temporary**. For **Input Title**, type “**Your**” and choose color number **191** from the color palette. Position it at x,y coordinates 125x by 415y.



### **Line 30**

In the **Type** field, select **Title**, select **KST Font FUTURA20**, and select **Permanent**. For **Input Title**, type “**Name Here**” and when asked **Choose a Color From Your Image?**, select **Yes**.

Sharp Boy says:  
By making this **Title** event **Permanent**, the two previous **Title** events will also become **Permanent**.

You now see a cross-shaped cursor with a box around it. As you move your cursor, you will see the box change color to reflect the color of the image behind it. Move the cursor to the word “**Your**” that was placed on the screen in the previous **Title** event. When the color of the box matches the color of the text on the screen, press **[Enter]**.

Now position the text at x,y coordinates 189x by 415y.

*Each time the user selects **Vars** from the opening **Menu** of the sequence, the numeric variable **#count** needs to be reset to zero. This is done with the **Var Modify** event in **Line 31**.*

## Lines 31 - 41

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Var		Modify		#count = 0		
b	Var		Modify		\$name = " "		
	Var		Input	E	\$name 12L 4T 341x 410y		
	Var		Modify		#count += 1		
	Title		FUTURA20	T	Hello, @105x 317y 130c		
	Title		FUTURA20	T	Number = @25x 213y 130c		
	Var		FUTURA20	T	Sname@182x 317y 191c		
	Var		FUTURA20	T	#count@150x 213y 191c		
	IfThen				#count -- 4		menu
	IfThen				\$name -- "exit"		menu
							b

## Line 31

In the **Type** field, select **Var**, then select **Modify**, and select **Numeric Var**. For **Input A Var Name : #**, enter “count”. For **Operator** select =, choose **Number**, and enter “0”. The **Parameters** field should read: **#count = 0**.

*Now we want to build another loop that asks users to type their name, tells them “Hello,” and displays the number of times they do this. We start by modifying the variable \$name so that each time users are asked to enter their name, the field is blank.*

## Line 32

Press [Enter] in the **Label** field and enter “b”. In the **Type** field, select **Var**, then select **Modify**, and select **String Var**. For **Input Var Name : \$** enter “name”. For **Operator** select =, choose **Literal String**, then for **Input Literal**, press the spacebar one time and then press [Enter]. The **Parameters** field should read: **\$name = “ ”**.

*The event in line 32 modifies the variable \$name to make it blank.*

*The next event is **Var Input**, which accepts an entry from the keyboard and stores it in a string var \$name. Each time users enter their name from the keyboard, the variable \$name is updated with each user's name. Later in the sequence, we will compare the contents of the variable with a known using an **IfThen** event and branch to one of two places, depending on the results of the **IfThen** test.*

### **Line 33**

In the **Type** field select **Var**, then select **Input**, and select **Erase after Displaying**. Select **String Var**, and for **Input A Var Name** : \$type" name". Set the maximum length to "12" and select 4 as the **Var Type** (Alphanumeric and Space). Now use the cursor keys or mouse to position \$name at x,y coordinates 341x by 410y. This positions a box for the user's name right next to the title "Enter Your Name Here."

*At the beginning of the sequence, you declared a numeric variable #count, with an initial value of zero. Now as part of this loop, we want to increment #count by one each time users enter their name. Later in the sequence, we will compare the value of the variable #count using an **IfThen** event, and branch to one of two places depending on the result of the **IfThen** test.*

### **Line 34**

In the **Type** field, select **Var**, select **Modify**, select **Numeric Var**, for **Input A Var Name** : \$, type "count". We want to add one to whatever the existing value is, so for **Test** select +=. Choose **Number**, and type "1". The **Parameters** field should now read: #count += 1.

*The above event takes whatever #count is currently equal to (=) and adds (+) one to it, then stores the new value in #count.*

*We now want to create two more **Title** events, one for "Hello" and one for "Number." When the sequence is run, "Hello" will appear on the screen next to the user's name. "Number" appears next to the number representing how many times a user has entered a name.*

**Line 35**

In the **Type** field, select **Title**, select **KST Font FUTURA20**, and select **Temporary**. Enter **“Hello”** for **Input Title**, and select color number **130**. Position the title at x,y coordinates 105x by 317y.

Sharp Boy says:

When the color you want to use is already on the screen, you can select to **Choose a Color From Your Image** and position the cross-shaped cursor over the desired color, then press **[Enter]**.

**Line 36**

In the **Type** field, select **Title**, select **KST Font FUTURA20**, and select **Temporary**. Enter **“Number”** for **Input Title**, and select color number **130**. Position the title at x,y coordinates 25x by 213y.

*Now we want to display on the screen the string variable \$name and the numeric variable #count next to “Hello” and “Number” respectively. This is accomplished by using a **Var Display** event for each variable.*

**Line 37**

In the **Type** field, select **Var**, select **Display**, select **KST Font FUTURA20**, and for **Display**, select **Temporary**. Select **String Var**, enter **“name”** and choose color number **191**. Position it at x,y coordinates 182x by 317y.

**Line 38**

In the **Type** field, select **Var**, select **Display**, select **KST Font FUTURA20**, and for **Display** select **Temporary**. Select **Numeric Var**, enter **“count”** and choose color number **191**, and position it at x,y coordinates 150x by 213y.

*We will now create two **IfThen** events. One compares the name entered from the **Var Input** event in Line 33. The other compares the number of times a name has been entered by testing the value of **#count** that is updated with the **Var Modify** event in Line 34. When either of these **IfThen** event tests become true, the sequence branches to the opening **Menu** of the sequence.*

**Line 39**

In the **Type** field, select **IfThen**, for **Var Type** select **Numeric**. Enter **“count”** for the **Var Label**, select **>=** for the **Test**, choose **Number**, and type **“4”**. Select **Don’t wait** and type **“menu”** in the **Next** field.

*The event in Line 39 tests the numeric variable #count to see if it is equal to four. If #count does equal four, TEMPRA Media Author branches to the opening Menu of the sequence. If #count does not equal four, the next line in the sequence is executed.*

### Line 40

In the **Type** field, select **IfThen**, for **Var Type** select **String**. Enter “name” for the **Var Label**, select **==** for the **Test**, choose **Literal String**, and type “exit”. Select **Don’t wait** and type “menu” in the **Next** field.



*The event in Line 40 tests the variable \$name to see if the name entered with the Var Input event in Line 33 is “exit.” If the variable \$name is “exit,” TEMPRA Media Author branches to the opening Menu of the sequence. If \$name is not “exit,” the next line in the sequence is executed.*

Sharp Boy says:

By placing several **IfThen** events together, you can test a single variable for a number of possible values.

*If neither of the IfThen event tests in Lines 39 and 40 are true, we want the sequence to branch back to the event where the users can again enter their name. We entered “b” in the Label field of Line 32, which is where we want to branch to. So we need a Next to match that Label.*

### Line 41

Move your cursor to the **Next** field and press [Enter]. Type “b” and press [Enter]. This creates an event line that does nothing but branch back to a label called “b.” Advance the Line Counter to Line 42 for the next event.

*To summarize the function of Lines 39-41, if #count does not equal four, and if \$name is not “exit,” we want to loop back to the point where the users enter their name. This point is identified with “b” in the Label field. When neither of the above IfThen tests are true, the users are again prompted to enter their name. When they have done this four times, or when they have entered “exit” as their name, TEMPRA Media Author branches back to the opening Menu of the sequence.*

The number four was used only as an example to show how TEMPRA Media Author can be used to keep track of how many times something happens in the sequence. We could just as easily have chosen two or twelve.

We choose the word “**exit**” to compare to the name entered by users. Again, this was simply to provide an example. TEMPRA Media Author can compare a known, in this case the word “**exit**”, with an unknown, by using a variable to hold the contents of the unknown, in this case the name entered by users.

### **Menu Item 2 — Dir**

Now we want to create the events for the **Dir Menu** selection. This series of events displays a **Menu** of the files stored in the TEMPRA Media Author **Screen** directory. The users can select one of the files that are listed, and depending on the type of file selected, TEMPRA Media Author branches to either a **Screen** or **Video** event and displays the image on the **Screen**. The sequence then returns to the opening **Menu**.

*We will start by creating a **TmOut NoKeyJMP** event. This event tells TEMPRA Media Author to branch to the opening **Menu** of the sequence if a selection from the **Dir** menu is not made within 15 seconds. You can vary this time if you like.*

*When we created the **Menu** in Lines 21 - 26, we entered “**dir**” as the **Next Label** for the **Dir** menu selection. The label we enter below must match exactly the label entered in Line 24, otherwise the sequence will not know where to branch to when **Dir** is selected from the **Menu**.*

### **Lines 42 - 44**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
dir	TmOut		NoKeyJMP			15 0/18	menu
	Var		Modify		\$path = &screen		
	Var		Modify		\$path += "\*.*)"		

### **Line 42**

Press [Enter] in the **Label** field and type “**dir**” (this matches the **Next** label entered in Line 24). Select **TmOut**, select **NoKeyJMP**, and enter “**15**” seconds. Type “**menu**” in the **Next** field.

*The event in Line 42 branches to the opening **Menu** if a selection is not made within 15 seconds.*

*At the beginning of the sequence, we declared a string variable named **\$path**, which holds the DOS pathname TEMPRA Media Author will use to find the image files. Now we want to modify that variable to hold the name of the directory and to mask the file types that are to be displayed. This is accomplished with two **Var Modify** events in Lines 43 and 44.*

### **Line 43**

In the **Type** field, select **Var**, select **Modify**, select **String Var**, and type “**path**” for the **Var Name**. For the **Operator** select = and scroll down to select **Screen Directory**.

*By using the system variable **Screen Directory**, TEMPRA Media Author will be able to find the image files, regardless of the directory in which they are stored.*

*Now we want to further modify the variable **\$path** to include a mask for file types to display.*

### **Line 44**

In the **Type** field, select **Var**, select **Modify**, select **String Var**, and type “**path**” for the **Var Name**. We want to add a mask for the file type to the existing value of the variable **\$path**, so for the **Operator** select +=. Choose **Literal String** and type “\\*.\*” (backslash asterisk period asterisk).

*By using the file mask “\\*.\*,” all files will be included. Therefore, this mask was not really necessary, but we wanted to show how the mask is used. If you wanted to display only PCX files, you would change the mask to read “\\*.PCX” and the **Dir** event would then only display PCX files.*

After you have entered Line 44, press [Alt]-V to display the TEMPRA Media Author variables table. The list of variables should include **\$path** = “C:\TEMPRA\PIC\\*.\*”, assuming TEMPRA Media Author is installed on drive C: in the TEMPRA directory and image files are stored in the PIC directory.

Next we create the **Dir** event. When **Dir** is selected from the opening **Menu**, users are shown a list of files located in their screen directory. This list appears in the form of a menu. When they select one of these files, the filename will be read into the variable **\$\_file**. The variable **\$path** that was modified in Line 43 tells **TEMPRA Media Author** the name of the directory (**&screen**) to display. This same variable **\$path** that was again modified in line 44, also tells **TEMPRA Media Author** which files (**\\*.\***) to display.

### Lines 45 - 51

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
	Dir	\$_file	\$path		239x 350y "Images"		
	IfThen				\$_file >- ".FLC"		vid
	IfThen				\$_file >- ".FLI"		vid
	IfThen				\$_file >- ".PCX"		scr
	IfThen				\$_file >- ".TGA"		scr
	IfThen				\$_file -- ""		menu
							dir

### Line 45

In the **Type** field, scroll down to select **Dir**. For **Input Var for File name \$**, the default is "**\_file**" (this matches the variable declared in Line 8 of the sequence). Press **[Enter]** to accept this default. For **Input Var with path and mask \$**, enter "**path**" (this is the variable that was modified in the above event Lines 43 and 44). For **Input DIR Title**, type "**Images**" and position the **Menu** in the center of the screen (239x by 350y).

The next series of **IfThen** events determine the type of image file users select from the **Images Menu** and branch to either a **Video** event to play **FLC** and **FLI** files or branch to a **Screen** event to display **PCX** and **TGA** files. If users select something other than one of these four types of files, **TEMPRA Media Author** will redisplay the **Images Menu**.

### Line 46

In the **Type** field, select **IfThen**. For **Var Type** select **String**. For **Input A Var Label** enter “\_file” (be sure to enter the underscore character) and for **Test** select **>=**. Choose **Literal String**, and for **Input Literal** type “.FLC” (be sure to enter the period). In the **Next** field, type “vid”.

*This event tests the variable \$file, which holds the name of the file selected from the Images Menu, against the literal string “.FLC.” The >= test means IF the selected file contains the literal string “.FLC,” THEN the test is true and TEMPRA Media Author branches to the label in the Next field, in this case, “vid.” If the test is not true, the next event in the sequence is executed.*

*Lines 47 - 49 perform the same function as Line 46, except that they test for different types of files.*

### Line 47

From the **Type** field, select **IfThen**. Select **String** as the **Var Type**, enter “\_file” as the **Var Label**, select **>=** as the **Test**, choose **Literal String**, enter “.FLI” for **Input Literal**, and enter “vid” in the **Next** field.

*Line 47 determines if the selected file is a FLI file and if so, branches to an event with “vid” in the Label field.*

### Line 48

From the **Type** field, select **IfThen**. Select **String** as the **Var Type**, enter “\_file” as the **Var Label**, select **>=** as the **Test**, choose **Literal String**, enter “.PCX” for **Input Literal**, and enter “scr” in the **Next** field.

*Line 48 determines if the selected file is a PCX file and if so, branches to an event with “scr” in the Label field.*

### Line 49

From the **Type** field, select **IfThen**. Select **String** as the **Var Type**, enter “\_file” as the **Var Label**, select **>=** as the **Test**, choose **Literal String**, enter “.TGA” for **Input Literal String**, and enter “scr” in the **Next** field.



Sharp Boy says:  
By placing several **IfThen** events together, you can test a single variable for a number of possible values.

*To summarize Lines 45 - 49, a menu of image files is displayed on the screen from which the users choose a file. The contents of the variable **\$Path** determine which files are displayed. When the users select a file from the menu, the filename is read into the variable **\$\_file**. The contents of **\$\_file** is then tested with a series of **IfThen** events in Lines 46 - 49. Each **IfThen** event tests for a different type of file. When one of the tests becomes true, **TEMPRA Media Author** branches to the event specified by the **Label** entered in the **Next** field.*

*Now we will provide a way for users to escape from the **Menu** of files. This will be accomplished with an **IfThen** event that tests to see if the **[Esc]** has been pressed.*

### Line 50

Select **IfThen** from the **Type** field, select **String** as the **Var Type**, enter “**\_file**” as the **Var Label**, select **==** for the **Test**, and select **Literal String**. When prompted to **Input Literal**, press the **[Enter]** key without entering anything from the keyboard. This enters a “**null**” as the literal value for which to test. In the **Next** field, enter “**menu**”. Your **Parameters** field should now read **\$\_file == “”**. Notice that there is no space between the quotation marks.

*This event returns users to the opening **Menu** of the sequence if they press the **[Esc]** key.*

*There may be files in the users’ screen directory other than the ones tested for in Lines 46 - 49. If the users select one of these other types of files, none of the **IfThen** event test will be true. Therefore, we want to redisplay the **Images Menu** when this occurs. This is done by simply putting a label in the **Next** field of Line 51.*

### Line 51

Now position the cursor in the **Next** field and press **[Enter]**. Type “**dir**” and press **[Enter]**. *This event branches to Line 42 of the sequence, which is where the events for the **Images Menu** begin.*

Before you can enter the next event, you must advance the Line Counter to Line 52.

*In Lines 46 - 49, we entered labels in the Next field of the IfThen events. Now we need to create the events for those events to branch to when the results of the IfThen tests are true.*

*The first one is a Screen event. When the results of the IfThen tests for Lines 48 or 49 become true, the sequence branches to Line 52, which we will give the label “scr” to match the one entered in the Next field of those events.*

### Lines 52 - 54

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
scr	Screen	\$_file	Lft_Rt	90		5 0/18	menu
vid	BColor		Quick		Solid 63c		
	Video	\$_file	AaPlay	95 T	161bx 329by 1R	2 0/18	menu

### Line 52

In the **Label** field, press [Enter], type “scr” and in the **Type** field, select **Screen**. When the list of files appear on the **Screen**, press [Esc]. *Instead of a specific filename being selected for this event, we want to read the Var Label, which holds the name of the file that is selected from the Images Menu.*

To enter this **Var Label**, position the cursor in the **File** field and press [F3]. Type “\$\_file” and press [Enter]. Be sure to include the dollar sign and the underscore character.

We now need to fill in the rest of the fields for this event. Move the cursor to the **Method** field, press [Enter], and select **Lft\_Rt**. Move the cursor to the **Sp/Dp** field, press [Enter], and change the speed to “90”. Move the cursor to the **Wait** field, press [Enter], select **Wait for Timeout**, and enter

“5” seconds. Move the cursor to the **Next** field, press [Enter], and type “menu”. Advance the Line Counter to enter the next event.

*This event reads the filename stored in the variable `$_file` and displays the TGA or PCX image on the screen. Then, the sequence branches to the opening **Menu** of the sequence.*

*When the results of the **IfThen** tests for Lines 46 or 47 become true, the sequence branches to Line 53, which we will give the label “vid” to match the one entered in the **Next** field of those events.*

### Line 53

Press [Enter] in the **Label** field, and type “vid”. In the **Type** field, select **BColor**, select **Quick**, and select color number **63** from the color palette.

### Line 54

In the **Type** field, select **Video**, select **AaPlay**, and when the list of files appears on the **Screen**, press [Esc]. Position the cursor in the **File** field and press [F3]. Type “`$_file`” and press [Enter]. Here we wanted to use the name of the file held in the variable `$_file`, rather than select a specific file.

In the **Sp/Dp** field change the speed to “95” and select **Temporary**. Position the cursor in the **Parameters** field and press [Enter]. You should then see the outline of a box on the screen. Move this box to the center of the screen (161x by 329y), press [Enter], and enter “1” for **Number of Repeats**.

In the **Wait** field, select **Wait for Timeout** and enter “2” seconds. Move to the **Next** field and enter “menu” as the label. Advance the Line Counter to enter the next event.

*To summarize what these events do: The **IfThen** events compare the **Var Label** `$_file` with the file selected from the **Images Menu**. Depending on which test is true, **TEMPRA Media Author** branches to a label “vid” or “scr.” When either of these two events have completed their function, **TEMPRA Media Author** returns to the opening **Menu** of the sequence.*

## Menu Item 3 — DBMS

This next series of events demonstrate the **DBMS** capabilities of **TEMPRA Media Author**. These events will open a database named “**IMAGES.DBF**,” read the information from each record into variable labels, test a label to determine the type of image file, and then branch to either a **Screen** event or a **Video** event, depending on the file type. The **Screen** or **Video** event will read another label to use as the filename, display the particular image on the **Screen**, and then start the whole process over.



Sharp Boy says:

To keep this tutorial simple, the database file **IMAGES** only has four records, with three fields in each record. Your database can have up to 128 fields, with a maximum record length of 4000 bytes.

After **TEMPRA Media Author** has displayed each image file that is listed in the database, the database closes and control returns to the opening **Menu** of the sequence.

At this time, you should refer to the **DBMS** chapter of this manual and review the contents of **IMAGES.DAT** and **IMAGES.SPC**. These are the data and specification files respectively for the database file **IMAGES.DBF**.

*In Line 25 we entered “**dbms**” as the **Next** label for the **DBMS** menu item. The label we enter below must match the one from Line 25. Then we will bring in a background color.*

### Lines 55 - 57

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
dbms	BColor		Quick		Solid 29c		
	DBMS	IMAGES	Open				
	DBMS		Goto		\$_dmax		

### Line 55

Press [Enter] in the **Label** field, and enter “**dbms**”. From the **Type** field, select **BColor**, select **Quick** as the **Method**, and choose color number **29** from the color palette.

Now we want to open the database and move the record pointer to the last record in the database.

**Line 56**

From the **Type** field, select **DBMS**, select **Open**, and select **IMAGES**.

**Line 57**

Again from the **Type** field, select **DBMS**, select **Goto**, select **\$ Var Label**, and enter “**\_dmax**”.

*The variable **\$\_dmax** is an internal database variable, which holds the record number of the last record in the database. **Goto \$\_dmax** tells TEMPRA Media Author to go to the last record.*

*The next event tells TEMPRA Media Author to read each field of the record into a separate var label. The first label is **\$drc01** and holds the contents of the first field of the record. The second label is **\$drc02** and holds the contents of the second field of the record. The third label is **\$drc03** and holds the contents of the third field of the record. The event needs a label because the sequence branches back to it from Line 64.*

**Lines 58 - 65**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
read	DBMS		Read		\$drc 1		
	IfThen				\$drc02 >= "FL"		dbvid
ifnot							dbscr
skip	Var		BLOCK30	T	\$drc01@72x 238y 191c		
	Var		BLOCK20	T	\$drc03@80x 194y 191c	2 0/18	
	IfThen				\$_drec == \$_itop		done
	DBMS		Skip		-1		read
done	DBMS		Close				menu

### Line 58

In the **Label** field, enter “read” and in the **Type** field, select **DBMS**. Select **Read**. For **Input Var Prefix Name:\$** enter “drc”. For **Input Starting Index: “1”** should be the default, so press [Enter].

*Next we create an **IfThen** event to test var label \$drc02. This is the var label that holds the filename extension. The contents will determine if TEMPRA Media Author branches to a **Video** in Line 68 or a **Screen** event in Line 66.*

### Line 59

In the **Type** field, select **IfThen**, for **Var Type** select **String**, for **Input A Var Label:\$** enter “drc02”. Select **>=** for the **Test**, choose **Literal String**, and for **Input Literal String type “FL”**. Enter “dbvid” in the **Next** field.

*Line 59 tests the var label \$drc02 to see if it contains the literal string “FL”. If the test is true, it means that either a **FLI** or **FLC** file is represented in the current record of the database, and if so, TEMPRA Media Author branches to the label “dbvid” in Line 68.*

*If the result of the **IfThen** test in Line 59 is not true, Line 60 will execute.*

### Line 60

In the **Label** field, enter “ifnot” and then move to the **Next** field and enter “dbscr”. Advance the **Line Counter** to enter the next event.

*If the **IfThen** test in Line 59 is not true, this means the file in the current record of the database is either a **PCX** or **TGA** file. In this case, Line 60 tells TEMPRA Media Author to branch to the label “dbscr” in Line 66.*

*After Lines 66 - 69 display the images on the screen, we want to do three things: display the name of the image file being shown, display the comments about the image, and test the current record number to see if we have reached the top record in the database. If we have not yet reached the top record, we want to decrement the record counter and read the next record. When we have reached the top record, we want to close the database and return to the opening **Menu**.*

---

*If this seems confusing, refer to Lines 56 and 57 and notice that after we opened the database, we moved the record counter to the last record and started reading each field of that record.*

*Line 61 needs a label because we branch back to it from Lines 67 or 69.*

### **Line 61**

Press [Enter] in the **Label** field and type “skip”. In the **Type** field, select **Var**, select **Display**, select **KST Font BLOCK30**, select **Temporary**, and select **String Var**. For **Input A Var Name:\$** enter “drc01” and choose color number **191** from the color palette. Position it at x,y coordinates 72x by 238y and select **Don't Wait**. *This event displays the name of the image file shown on the screen.*

### **Line 62**

In the **Type** field, select **Var**, select **Display**, select **KST Font BLOCK20**, select **Temporary**, and select **String Var**. For **Input A Var Name:\$** enter “drc03” and choose color number **191** from the color palette. Position it at x,y coordinates 80x by 194y. Select **Wait for Timeout** and enter “2” seconds. *This event displays a comment about the image file shown on the screen.*

*The event in Line 63 compares the record number for the image that was last displayed with the top record number in the database. When these two record numbers match, we branch to the label “done” in Line 65, which closes the database and branches to the opening Menu.*

### **Line 63**

In the **Type** field, select **IfThen**. For **Var Type** select **String**, for **Input A Var Label:\$** enter “\_drec” (be sure to enter the underscore character). For **Test** select **==**, choose **String Var**, and for **Input A Var Label:\$** enter “\_itop” (notice the underscore character). Select **Don't wait** and enter “done” in the **Next** field.

*If \$\_drec does not equal \$\_top, we want to decrement the record pointer by one and then read the fields of the next record into the variable labels.*

**Line 64**

In the **Type** field, select **DBMS**, select **Skip**, and enter “-1” (we use minus one because we started with the last record in the database and are working toward the top). Select **Don’t wait**, and enter “**read**” in the **Next** field.

*The event in Line 64 decrements the record pointer by one and branches to Line 58, which matches the “read” label.*

*If the results of the IfThen test in Line 63 are true, \$\_drec does equal \$\_top, and the sequence branches to Line 65, which closes the database and returns to the opening Menu.*

**Line 65**

Press [Enter] in the **Label** field and type “done”. In the **Type** field, select **DBMS**, select **Close**, and enter “**menu**” in the **Next** field. *This event closes the database and branches to the opening Menu of the sequence.*

*Now that all the counting and testing and pointing and skipping is finished, we need to create the events that will display the images listed in the database. We need a background color and Screen event to display PCX or TGA files and these events need the label “dbscr” to match the label in the Next field of Line 60. We also need a background color and Video event to play FLI or FLC files and these events need the label “dbvid” to match the label in the Next field of Line 59.*

**Lines 66 - 69**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
dbscr	BColor		Quick		Solid 7c		
	Screen	\$drc01	Quick		45bx 220by 364ex 420ey	2 0/10	skip
dbvid	BColor		Quick		Solid 7c		
	Video	\$drc01	AaPlay	95 T	45bx 420by 1R	2 /010	skip

**Line 66**

Press [Enter] in the **Label** field, type “dbscr” and select **BColor** from the **Type** field. Select **Quick**, and choose color number 7 from the color palette. Select **Don’t wait** and press [Enter] in the **Next** field.

*We don’t want a specific filename for the **Screen** event, but instead want the file represented in the current record of the database. The name of this file is identified in the variable \$drc01.*

**Line 67**

In the **Type** field, select **Screen**. When the list of files pops up on the screen, press [Esc], move the cursor to the **File** field and press [F3], and enter “\$drc01”. Move to the **Method** field and select **Quick**.



Sharp Boys says:  
When positioning images on the screen, remember to press [Ctrl]-X to display the x,y coordinates.

Move to the **Parameters** field and select **Scaled**. A message in the upper left corner of the screen directs you to place the cursor at the lower left hand corner and press [Enter]. Position the cursor at x,y coordinates 45x by 220y and press [Enter]. Now when you move the cursor, the box grows. Expand the box to x,y coordinates 364ex by 420ey and press [Enter].

Select **Wait for Timeout**, enter “2” seconds, and enter “skip” in the **Next** field. Now advance the **Line Counter** to 68 for the next event.

*We entered the label “skip” in the **Next** field so that after the image from the database is displayed on the screen, the sequence branches to the event that displays the filename and comments, decrements the record counter and reads the next record.*

**Line 68**

Press [Enter] in the **Label** field, type “dbvid” and from the **Type** field, select **BColor**. Select **Quick**, and choose color number 7. Select **Don’t wait** and press [Enter] in the **Next** field.

*We don’t want a specific filename for the **Video** event, but instead want the file represented in the current record of the database. The name of this file is identified in the variable \$drc01.*

**Line 69**

In the **Type** field, select **Video** and select **AaPlay**. When the list of files pops up on the screen, press [Esc]. Move the cursor to the **File** field, press [F3], and enter “\$drc01”. In the **Sp/Dp** field change the speed to “95” and select **Temporary**.

Move the cursor to the **Parameter** field and press [Enter]. You should see the outline of a box on the screen. Move the cursor to position the box at x,y coordinates 45bx by 420by, and press [Enter]. For number of repeats, “1” is the default, so press [Enter]. Select **Wait for Timeout**, enter “2” seconds, and enter “skip” in the **Next** field. Now advance the Line Counter to 70 for the next event.

*We entered the label “skip” in the Next field so that after the image from the database is displayed on the screen, the sequence branches to the event that displays the filename and comments, decrements the record counter and reads the next record.*

Now all that’s left to complete this sequence is a nice, orderly way to exit. This will be done with **BColor**, **Title**, and **Text** events.

**Lines 70 - 76**

Label	Type	File	Method	Sp/Dp	Parameters	Wait for	Next
exit	Bcolor		Quick		Sol1c 0c		
	BColor		Hrz_Out	10	Solid 131c		
	Title		TIMES25	T	Thank You@218x 322y 191c	1 0/18	
	Title		TIMES25	T	For Using@225x 233y 191c	1 0/18	
	Text	MA	TIMES25	T 001	142x 265y 102c 14w 3h nfm	3 0/18	
	BColor		Hrz_In	10	Solid 98c		
	BColor		Hrz_Out	10	Solid 0c		

**Line 70**

Press [Enter] in the **Label** field. Type “exit” to match the label entered in the **Next** field of Line 26 of the **Menu** event. Select **BColor**, select **Quick**, and choose color number **0** from the color palette.

**Line 71**

From the **Type** field, select **BColor**, select **Hz\_Out**, change the speed to “10” and choose color number **131** from the color palette. Select **Don't wait** and press [Enter] in the **Next** field.

**Line 72**

From the **Type** field, select **Title**, select **KST Font TIMES25**, and select **Temporary**. For **Input Title** type “Thank You” and choose color number **191** from the color palette. Position the **Title** at x,y coordinates 218x by 322y. Select **Wait for Timeout** and enter “1” for the number of seconds, and press [Enter] in the **Next** field.

**Line 73**

From the **Type**, field select **Title**, select **KST Font TIMES25**, and select **Temporary**. For **Input Title** type “For Using” and choose color number **191** from the color palette. Position the **Title** at x,y coordinates 225x by 233y. Select **Wait for Timeout** and enter “1” for the number of seconds, and press [Enter] in the **Next** field.

**Line 74**

From the **Type** field, select **Text**, select **MA.TXT**, select **KST Font TIMES25**, and select **Temporary**. Move your cursor to point to the first line of **Text** and press [Enter]. Answer **No** to **Allow User to Operate?**, answer **Yes** to **Draw Frame?** and select **Middle** for **Justify**. Choose color number **102** from the color palette.

Now use your mouse to position the frame around the text and press [Enter]. You should then see a large gray box on the screen. Position the box at x,y coordinates 142x by 265y. Select **Wait for Timeout** and enter “3” for the number of seconds, and press [Enter] in the **Next** field.

### Line 75

From the **Type** field, select **BColor**, select **Hrz\_In**, set the speed to “**10**” and choose color number **98** from the color palette. Select **Don't Wait** and press **[Enter]** in the **Next** field.

### Line 76

From the **Type** field, select **BColor**, select **Hrz\_Out**, set the speed to “**10**” and choose color number **0** from the color palette. Select **Don't Wait** and press **[Enter]** in the **Next** field.

## ***Success!***



Now that the sequence is complete, point to the **Disk Button** on the left side of the **Menu bar** and press **[Enter]**. Select **Save Sequence**, and select **Overwrite File**. Click on the **Line Counter** to position the sequence at **Line 1**, then press **[F6]** to run the sequence from the beginning.

Sharp Boy says:

Thanks for taking the time to do this tutorial. You should now have a better understanding of the power of **TEMPRA Media Author**.

---

# ***Index***

---

# Index

## A

APlay, Video, 35-10  
 Animation, 35-9  
 Answers, IfThen, 21-2  
 Append, DBMS, 11-3  
 As Is, Screen Event, 26-2  
 Ask For Label, 4-4  
 Audio Board Type, Appendix H-5  
 Audio Button, 2-6  
 Audio Event, 5-2, 6-1
 

- Chapter Tutorial, 6-3
- Other Fields, 6-3
- Play, 6-1
- Playasnc, 6-2
- Quiet, 6-2
- Record, 6-2
- Type, 6-1

 Audio INT/IMA/IRG, Appendix H-7  
 Audio Track 1, Vdisc, 36-3  
 Audio Track, 2, Vdisc, 36-3  
 Audio1 & Audio2, Vdisc, 36-3  
 AUTOEXEC.BAT, 1-10, Appendix H-2  
 AutoOff, Global, 19-2  
 Autoset, Global, 19-2

## B

BackOff, Global, 19-3  
 BackOn, Global, 19-3  
 BColor Event, 5-2
 

- Chapter Example, 7-4
- Method, 7-1
- Other Fields, 7-4
- Parameters, 7-3
- Sp/Dp, 7-3
- Type, 7-1

 BEEP Environmental Variable, Appendix H-15  
 Black & White Video, VidBW, 35-18  
 BLAST\_BUF Environmental Variable, Appendix H-15  
 Block, 3-3
 

- Copy, 3-4
- Delete, 3-4
- Load, 3-5

Marking, 3-3

Move, 3-4

Operations, 3-3

Save, 3-5

Unmark, 3-5

Border Background, Global, 19-4

Border Foreground, Global, 19-4

Button Default Cursor, 8-2

Button Event, 5-2, 8-1

Chapter Tutorial, 8-4

Default Cursor, 8-2

File, 8-1

Labels, 8-3

Method, 8-2

Next, 8-3

Other Fields, 8-4

Parameters, 8-2

Type, 8-1

Buttons, Audio, 2-6

Clear, 2-3

Delete Line, 2-8

Disk, 2-3

Exit, 2-2

Help, 2-6

Insert Line, 2-8

Menu Position, 2-7

Path, 2-5

Play Line, 2-7

Play Sequence, 2-8

Remove Menu, 2-2

TEMPRA, 2-5

Word, 2-6

Work, 2-6

## C

CD-ROM Event, 5-2, 9-1

Chapter Example, 9-4

Continue, 9-2

Eject, 9-3

Load, 9-2

Method, 9-1

Other Fields, 9-3

Parameters, 9-3

Pause, 9-2

- Play, 9-1
- Playrang, 9-1
  - Parameters, 9-3
- Seek, 9-2
  - Parameters, 9-3
- Stop, 9-2
- Type, 9-1
- CEG Files, 19-4, 35-7
- CEGMODE, Video, 35-7
- Chain Event, 5-2, 10-1
  - Chapter Example, 10-2
  - File, 10-2
  - Other Fields, 10-2
  - Type, 10-1
  - Wait For, 10-2
- Clear Button, 2-3
- ClockJMP, TmOut, 31-3
- ClockOff, Video, 35-8
- ClockOn, Video, 35-8
- ClockSUB, TmOut, 31-4
- Close, DBMS, 11-3
- CLP Files, 19-4
- Color Menu, 7-3
- Color, Text, 29-3
- Colors
  - Global, 19-2
  - Config Program, Appendix H-4
- Commands, Serial I/O, Appendix C-2
- Common Fields, 4-1
- Config, PLComm, 25-2
- CONFIG.SYS, 1-10, 28-1, Appendix H-1
- CONFIGMA, 34-1, Appendix H-2, H-19
- Configuration Program, Appendix H-2
- Continue
  - CD-ROM, 9-2
  - Midi, 23-2
- Create New Titles, 30-2
- CycleCol, Video, 35-8

## **D**

- DBF Files, 11-1
- DBMS Event, 5-2, 11-1
  - Append, 11-3
  - Close, 11-3
  - Delete, 11-3
  - Examples, 11-16, 11-17, 11-18
  - FieldNam, 11-3
    - Parameters Example, 11-8
  - File, 11-1

- Goto, 11-3
  - Parameters Example, 11-8
- Index, 11-4
  - File, 11-1
- IndexOff, 11-4
- Internal Database Variables, 11-18
- Lock, 11-4
- MakeDBF, 11-4
  - File, 11-2
  - Parameters Example, 11-9
- MakeNDX, 11-5
  - File, 11-2
  - Parameters Example, 11-10
- Memo2txt, 11-5
  - File, 11-2
  - Parameters Example, 11-12, 11-15
- Method, 11-2
- Next, 11-18
- Open, 11-5
  - File, 11-2
- Other Fields, 11-18
- Pack, 11-5
- Parameters, 11-8
- Read, 11-6
  - Parameters Example, 11-12
- ReadFld, 11-6
  - Parameters Example, 11-12
- ReIndex, 11-6
  - File, 11-2
- Rlock, 11-6
- Seek, 11-6
  - Parameters Example, 11-13
- Skip, 11-7
  - Parameters Example, 11-13
- Txt2memo, 11-7
  - File, 11-2
  - Parameters Example, 11-14
- Type, 11-1
- Undelete, 11-7
- Unlock, 11-8
- Write, 11-7
  - Parameters Example, 11-14
- WriteFld, 11-7
  - Parameters Example, 11-14
- DBT Files, 11-2
- Declare, Var, 33-2, 33-4
- Delay, Global, 19-2
- Delete
  - DBMS, 11-3
  - File, 2-4

- Delete Line Button, 2-8
- Demo Builder, 2-5, 2-15
- Dir Event, 5-3, 12-1
  - Chapter Example, 12-3
  - File, 12-2
  - Method, 12-2
  - Other Fields, 12-3
  - Parameters, 12-3
  - Sp/Dp, 12-3
  - Type, 12-1
- Disk Button, 2-3
- Display, Sp/Dp field, 2-10, 4-5
- Distributable Presentations, 2-15
- Do Serial I/O, 4-4, Appendix C-1
- Don't Wait, 4-3
- Draw Event, 5-3, 13-1
  - Box, 13-2
    - Parameters, 13-3
  - Chapter Example, 13-4
  - Circle, 13-2
    - Parameters, 13-2
  - FBox, 13-2
    - Parameters, 13-3
  - FCircle, 13-2
    - Parameters, 13-3
  - Freehand, 13-2
  - Line, 13-1
    - Parameters, 13-2
  - Method, 13-1
  - Other Fields, 13-3
  - Parameters, 13-2
  - Set Brush, 13-2
    - Parameters, 13-3
  - Type, 13-1
- DropOff, Global, 19-2
- Dropshad, Global, 19-2

## E

- Editing Titles, 30-3
- EDITOREnvironmental Variable, Appendix H-16, H-22
- 80Butn Event, 5-3, 14-1
  - Chapter Example, 14-4
  - Method, 14-1
  - Next, 14-3
  - Other Fields, 14-3
  - Type, 14-1
  - Parameters, 14-2
  - Sp/Dp, 14-2

- 80Disp, Var, 33-3, 33-9
- 80Input, Var, 33-3, 33-9, 33-10
- 80Menu Event, 5-3, 15-1
  - Chapter Example, 15-4
  - Editing with Prompts, 15-3
  - Method, 15-1
  - Next, 15-3
  - Other Fields, 15-3
  - Parameters, 15-2
  - Sp/Dp, 15-2
  - Type, 15-1
- 80Mode Event, 5-3, 16-1
  - Chapter Example, 16-2
  - Method, 16-1
  - Other Fields, 16-2
  - Type, 16-1
- 80Text Event, 5-3, 17-1
  - Chapter Example, 17-3
  - File, 17-1
  - Other Fields, 17-3
  - Parameters, 17-2
  - Sp/Dp, 17-1
  - Type, 17-1
- 80Win Event, 5-3, 18-1
  - Chapter Example, 18-2
  - File, 18-1
  - Other Fields, 18-2
  - Parameters, 18-2
  - Sp/Dp, 18-2
  - Type, 18-1
- Eject, CD-ROM, 9-3
- Environmental Variables, 1-11, Appendix H-15
  - BEEP, Appendix H-15
  - BLAST\_BUF, Appendix H-15, H-23
  - EDITOR, Appendix H-15, H-22
  - MIDI\_BUF, Appendix H-16
  - MTBAUD, Appendix H-16
  - PROGX, Appendix H-16
  - TMP, Appendix H-16
  - VDISC\_BAUD, Appendix H-16
  - VMP\_FORMAT, Appendix H-16
  - VVDAC, Appendix H-17, H-20
  - VVLINES, Appendix H-17
  - VVMODE, Appendix H-17
  - VVPAINT, Appendix H-17
  - VVPIC, Appendix H-17
  - VVSHARE, Appendix H-17
  - VVSPRITE\_BUF, Appendix H-18
  - VV\_VARK, Appendix H-18

- Erase after Displaying, 2-10
- Error Messages, Appendix H-20
- Event Controller, 2-7
- Event Line, 2-1
- Event Types, 5-1
- Events, Making, 2-12
- Exit Button, 2-2
- Extended Capability, Var, 33-14
- External Program, 2-7

## F

- FadeIN/FadeOUT, Special Info, 35-5
- FadeIN, Video, 35-4
- FadeOUT, Video, 35-5
- FieldNam, DBMS, 11-3, 11-8
- Fields
  - File, 2-9
  - Label, 2-9, 4-1, 4-2
  - Method, 2-9
  - Next, 2-11, 4-1, 4-2
  - Parameters, 2-11
  - Sp/Dp, 2-10, 4-4
  - Type, 2-9, 4-1
  - Wait For, 2-11, 4-3
- File Field, 2-9
- FLC Files, 35-1, 35-8, 35-9
- FLI Files, 35-1, 35-8, 35-9
- Font, Default, 19-2, 29-2
- Font, Title, 30-1
- Fonts, Available, Appendix G-1
- Fonts, Global, 19-2
- Frame Display, Vdisc, 36-2, 36-3
- Framed/Not Framed, Text, 29-3
- Freq, Music, 24-3, 24-5
- Freqfile, Music, 24-3, 24-5

## G

- Get Frame, Vdisc, 36-2, 36-3
- Global Event, 5-3, 19-1
  - AutoOff, 19-2
  - Autoset, 19-2
  - BackOff, 19-3
  - BackOn, 19-3
  - Chapter Example, 19-6
  - Colors, 19-2
    - Parameters, 19-4
  - Delay, 19-2
    - Parameters, 19-5

- DropOff, 19-2
- Dropshad, 19-2
  - Parameters, 19-5
- File, 19-1
- Font, 19-2
- Method, 19-2
- Other Fields, 19-6
- Parameters, 19-4
- PIC\_CEG, 19-4
- PIC\_PCX, 19-4
- PIC\_TGA, 19-4
- QuickOff, 19-3
- QuickOn, 19-3
- ResetAll, 19-2
- ScaleOff, 19-3
- ScaleOn, 19-3
- SPR\_CLP, 19-4
- SPR\_PCC, 19-4
- SPR\_WIN, 19-4
- Style2D, 19-3
- Style3D, 19-3
- StyleTL, 19-4
- Type, 19-1
- Gosub Event, 5-4, 20-1
  - Chapter Example, 20-3
  - File, 20-2
  - Next, 20-2
  - Other Fields, 20-2
  - Type, 20-1
- Goto, DBMS, 11-3, 11-8
- Graphic User Interface (GUI), 2-2

## H

- Help, 2-5
- Help Button, 2-6
- Highlight Background, Global, 19-5
- Highlight Foreground, Global, 19-5
- Hot Keys, Runtime, Appendix A-3

## I

- IfThen Event, 5-4, 21-1
  - Answers, 21-2
  - Chapter Example, 21-5
  - Next, 21-4
  - Numeric Variable, 21-3
  - Other Fields, 21-4
  - Parameters, 21-1
  - String Variable, 21-3

- Type, 21-1
- Variables, 21-2
- Index, DBMS, 11-1, 11-4
- IndexOff, DBMS, 11-4
- Infowindow, 35-19
  - Infowind, 35-19
- Input Parameter, Numeric, Var, 33-8
- Input Parameters, String, Var, 33-7
- Input, Var, 33-2
- InsEdit1, VCR, 34-4, 34-5
- Insert Line Button, 2-8
- Installation, 1-7
- Internal Database Variables, DBMS, 11-18
- Introduction, 1-1
- Inverse Video, VidInv, 35-18

**J**

- Joystick Parameters, Appendix H-5
- Justify, Text, 29-4

**K**

- Keyboard Functions, Appendix A-1
- Keyboard Return Codes, Appendix F-1
- Keyboard, Using the, 1-12
- KeyJMP, TmOut, 31-2
- KeySUB, TmOut, 31-3
- Kill, Music, 24-3, 24-4

**L**

- Label Field, 2-9, 4-1, 4-2
- Line
  - Delete, 2-8
  - Play, 2-7
  - Sample, 2-13
- Line Counter, 2-8
- Line Scroll Bar, 2-8
- Load
  - CD-ROM, 9-2
  - Parameter, Var, 33-8
  - Sequence, 2-3
  - Var, 33-3, 33-9
- LoadPal, Video, 35-2
- Lock, DBMS, 11-4
- Loop, Music, 24-4, 24-5

**M**

- MA.CFG, Appendix H-2

- MakeDBF, DBMS, 11-2, 11-4, 11-9
- MakeNDX, DBMS, 11-2, 11-5, 11-10
- Making Events, 2-12
- Memory, Appendix H-18
  - EMS, Appendix H-20, H-21
- Memo2txt, DBMS, 11-2, 11-12, 11-15
- Menu Bar, 2-2
- Menu Conventions, 2-1
- Menu Event, 5-4, 22-1
  - Chapter Tutorial, 22-4
  - Method, 22-1
  - Next, 22-3
  - Other Fields, 22-4
  - Parameters, 22-2
  - Type, 22-1
- Menu Position Button, 2-7
- Method Field, 2-9
- MID Files, 23-1
- MIDI\_BUF, Environmental Variable, Appendix H-16
- Midi Event, 5-4, 23-1
  - Chapter Example, 23-2
  - Continue, 23-2
  - File, 23-1
  - Information, Appendix H-10
  - Method, 23-2
  - Other Fields, 23-2
  - Play, 23-1, 23-2
  - Quiet, 23-2
  - Stop, 23-2
  - Type, 23-1
- Modify, Var, 33-2, 33-4, 33-5
- Motion Video, 35-15
- Mouse, Using the, 1-12
- MTBAUD Environmental Variable, Appendix H-16
- MUS Files, 24-1, 24-2, 24-3
- Music, Creating, Appendix D-1
  - Example, Appendix D-4
- Music Event, 5-5, 24-1
  - Chapter Example Sequence, 24-1, 24-7
  - File, 24-2
  - Freq, 24-3, 24-5
  - Freqfile, 24-3, 24-5
  - Kill, 24-3, 24-4
  - Loop, 24-4, 24-5
  - Method, 24-2
  - Other Fields, 24-6
  - Parameters, 24-4
  - Play, 24-2, 24-4

- Playfile, 24-2, 24-4
- Speaker, 24-4, 24-5
- Tone, 24-3, 24-5
- Tonefile, 24-3, 24-5
- Type, 24-2

## N

- NDX Files, 11-1
- New Sequence, 2-3
- Next Field, 2-11, 4-1, 4-2
- NoKeyJMP, TmOut, 31-2
- NoKeySUB, TmOut, 31-2
- Numeric Variable, IfThen, 21-3

## O

- Open, DBMS, 11-2, 11-5

## P

- Pack, DBMS, 11-5
- Parameters, Config Program, Appendix H-5
- Parameters Field, 2-11
- Path Button, 2-5
- Pause
  - CD-ROM, 9-2
  - VCR, 34-3
- PCCFiles, 19-4, 27-1, Appendix H-20, H-21, H-22
- PCX Files, 19-4, 26-1, 35-4
- Permanent events, 2-10
- PIC\_CEG, Global, 19-4
- PIC\_PCX, Global, 19-4
- PIC\_TGA, Global, 19-4
- Pioneer 6000 & 8000, 36-1
- Play
  - Audio, 6-1
  - Backward, Vdisc, 36-2
  - CD-ROM, 9-1
  - Forward, Vdisc, 36-2
  - Midi, 23-1, 23-2
  - Music, 24-2, 24-4
  - VCR, 34-2
- Play Line Button, 2-7
- Play Segment, VCR, 34-3
  - Parameter, 34-5
- Play Sequence Button, 2-8
- Playasnc, Audio, 6-2
- Playfile, Music, 24-2, 24-4
- Playrang, CD-ROM, 9-1

- Playseg, Vdisc, 36-2, 36-3
- PlaysegP, Vdisc, 36-2, 36-3
- PLComm Event, 5-5, 25-1
  - Chapter Example, 25-6
  - Config, 25-2
  - Method, 25-2
  - Next, 25-6
  - ReadCR, 25-3
  - ReadMem, 25-2
  - ReadVar, 25-2
  - Trap, 25-4
  - TrapJMPM, 25-5
  - TrapJMPP, 25-6
  - TrapJMPV, 25-5
  - TrapOFF, 25-6
  - TrapSUBM, 25-4
  - TrapSUBR, 25-5
  - TrapSUBV, 25-4
  - Type, 25-1
  - Version, 25-3
- Position, Text, 29-4
- Positioned, Screen Event, 26-2
- Presentation, Sample Line, 2-13
- Print Sequence, 2-4
- PROGX Environmental Variable, Appendix H-16
- PTH Files, 12-2
- PWin Utility, 18-1, Appendix E-1

## Q

- QuickOff, Global, 19-3
- QuickOn, Global, 19-3
- Quiet
  - Audio, 6-2
  - Midi, 23-2
- Quit, 2-5

## R

- Read
  - DBMS, 11-6, 11-12
  - Var, 33-3, 33-11
- ReadCR, PLComm, 25-3
- ReadFld, DBMS, 11-6, 11-12
- ReadMem, PLComm, 25-2
- ReadVar, PLComm, 25-2
- Record
  - Audio, 6-2
  - VCR, 34-4

ReIndex, DBMS, 11-2, 11-6  
 Remove Menu Button, 2-2  
 Rename Sequence, 2-4  
 Reserved Variable Names, Var, 33-14  
 ResetAll, Global, 19-2  
 ResetPal, 35-3  
 Rewind, VCR, 34-2  
 Rlock, DBMS, 11-6  
 RS232C Serial Connection, 36-1

## S

Sample Line, 2-13

Save

- Picture, 2-4
- Sequence, 2-4
- Var, 33-3, 33-9

SavePic, Video, 35-4

Scaled, Screen Event, 26-2

ScaleOff, Global, 19-3

ScaleOn, Global, 19-3

Screen Event, 5-5, 26-1

- As Is, 26-2
- Chapter Tutorial, 26-3
- File, 26-1
- Method, 26-2
- Other Fields, 26-3
- Parameters, 26-2
- Positioned, 26-2
- Scaled, 26-2
- Type, 26-1

Scrollup, Video, 35-11

Search

- VCR, 34-3, 34-4
- Vdisc, 36-2, 36-3

Seek

- CD-ROM, 9-2, 9-3
- DBMS, 11-6, 11-13

Send Contents, Vdisc, 36-3, 36-4

Sequence, 2-2

- Load, 2-3
- New, 2-3
- Play, 2-8
- Print, 2-4
- Rename, 2-4
- Save, 2-4

Serial I/O, 4-4, Appendix C-1

- Commands, Appendix C-2
- Get Command, Appendix C-5
- Initialize Command, Appendix C-2

Parallel Transmission, Appendix C-6

Reading from Text File, Appendix C-6

Receive Command, Appendix C-4

Send Command, Appendix C-4

SetMode, Video, 35-3

Skip, DBMS, 11-7, 11-13

Sony 2000, 36-1

Sony VBox, 34-1

Sony VDeck, 34-1

Sony VISCA, 34-1

Sp/Dp Field, 2-10, 4-4

SPC Files, 11-4

Speaker, Music, 24-4, 24-5

Speed, 2-10, 4-4

Split, Var, 33-3, 33-12

SPR\_CLP, Global, 19-4

SPR\_PCC, Global, 19-4

SPR\_WIN, Global, 19-4

Sprite Event, 5-5, 27-1

- Chapter Tutorial, 27-4

- File, 27-1

- Method, 27-2

- Other Fields, 27-3

- Parameters, 27-2

- Scale, 27-2

- Simple, 27-2

- Type, 27-1

- Wipe In, 27-2

Sprites, Creating, 27-3, 27-4

Sprites, Information, Appendix H-20

Sprites, Special Features of, 27-3

Start up Options, Appendix H-9

Step Backward, Vdisc, 36-2

Step Forward, Vdisc, 36-2

Stop

- CD-ROM, 9-2

- Midi, 23-2

- VCR, 34-2

- Vdisc, 36-2

String Variable, IfThen, 21-3

Style2D, Global, 19-3

Style3D, Global, 19-3

StyleTL, Global, 19-4

System Event, 5-5, 28-1

- Chapter Example, 28-2

- Method, 28-1

- Other Fields, 28-2

- Parameters, 28-2

- Type, 28-1

System Requirements, 1-6

**T**

Technical Information, Appendix H-1  
 Technical Support, 1-7  
 Temporary Events, 2-10  
 TEMPRA Button, 2-5  
 Terminology, 2-1  
 Text Background, Global, 19-4  
 Text Event, 5-6, 29-1  
     Allow User to Operate? menu, 29-2, 29-3  
     Chapter Tutorial, 29-5  
     Color, 29-3  
     File, 29-1  
     Framed or Not Framed, 29-3  
     Justify, 29-4  
     Method, 29-2  
     Operate Frame, 29-3  
     Parameters, 29-2  
     Position, 29-4  
     Sp/Dp, 29-2  
     Type, 29-1  
     Width/Height, 29-4  
 Text Foreground, Global, 19-4  
 TEXTEDIT, 29-1  
     Information, Appendix B-1  
     Installation, 1-9  
 TGA Files, 19-4, 26-1, 35-4  
 TI Key Option, Appendix H-5  
 Ticker, Video, 35-13  
 TimedJMP, TmOut, 31-3  
 TimedSUB, TmOut, 31-3  
 Title Box, 2-3  
 Title Event, 5-6, 30-1  
     Chapter Tutorial, 30-4  
     Create New Titles, 30-2  
     Editing Titles, 30-3  
     Font, 30-1  
     Method, 30-1  
     Other Fields, 30-3  
     Parameters, 30-2  
     Type, 30-1  
 TmOut Event, 5-6, 31-1  
     Chapter Example, 31-4  
     ClockJMP, 31-3  
     ClockSUB, 31-4  
     KeyJMP, 31-2  
     KeySUB, 31-3  
     Method, 31-2  
     NoKeyJMP, 31-2

NoKeySUB, 31-2  
 Other Fields, 31-4  
 TimedJMP, 31-3  
 TimedSUB, 31-3  
 Type, 31-1  
     Wait For, 31-4  
 TMP Environmental Variable, Appendix H-16  
 Tone, Music, 24-3, 24-5  
 Tonefile, Music, 24-3, 24-5  
 Toucans, TEMPRA, 1-3  
 Touch Screen, Appendix H-6  
     Indicator, Appendix H-7  
     Information, Appendix H-11  
     Serial Port, Appendix H-7  
 Track Event, 5-7, 32-1  
     Chapter Example, 32-2  
     Method, 32-2  
     Other Fields, 32-2  
     Type, 32-1  
 Transparent Color, VidTcol, 35-17  
 Trap, PLComm, 25-4  
 TrapJMPM, PLComm, 25-5  
 TrapJMPPR, PLComm, 25-6  
 TrapJMPV, PLComm, 25-5  
 TrapOFF, PLComm, 25-6  
 TrapSUBM, PLComm, 25-4  
 TrapSUBR, PLComm, 25-5  
 TrapSUBV, PLComm, 25-4  
 TRK Files, 32-1  
 Troubleshooting, Appendix H-18  
 Tutorial, TEMPRA Media Author, Appendix  
     J-1  
     Tutorial, TEMPRA SHOW, Appendix I-1  
 TXT Files, 12-2, 29-1, 35-11, 35-13  
 Txt2memo, DBMS, 11-2, 11-7, 11-14  
 Type Field, 2-9, 4-1

**U**

Undelete, DBMS, 11-7  
 Unlock, DBMS, 11-8

**V**

Var Event, 5-7, 33-1  
     Declare, 33-2  
         Parameters, 33-4  
     Display, 33-2  
         Parameters, 33-6  
     80Disp, 33-3  
         Parameters, 33-9

- 80Input, 33-3
  - Parameters, String, 33-10
  - Parameters, Numeric, 33-11
- Input, 33-2
  - Parameters, Numeric, 33-8
  - Parameters, String, 33-7
- Load, 33-3
  - Parameters, 33-9
- Method, 33-2
- Modify, 33-2
  - Parameters, Numeric, 33-4
  - Parameters, String, 33-5
- Next, 33-14
- Parameters, 33-4
- Read, 33-3
  - Parameters, 33-11
- Save, 33-3
  - Parameters, 33-9
- Sp/Dp, 33-4
- Split, 33-3
  - Parameters, 33-12
- Type, 33-2
- Write, 33-3
  - Parameters, 33-13
- Var Extended Capability, 33-14
- Var Reserved Variable Names, 33-15
- Variables, 33-1
  - IfThen, 21-2
  - Internal Database, DBMS, 11-18
  - Reading & Writing with Text Files, 33-17
  - Seeing the Values of, 33-17
  - Series of, 33-17
  - Using Successfully, 33-16
  - Using the Split Method, 33-18
  - Using the Write Method, 33-19
  - Using the Read Method, 33-18
  - Using to Get Files, 33-16
- VCR Event, 5-7, 34-1
  - Chapter Examples, 34-6
  - File, 34-2
  - InsEdit1, 34-4
    - Parameters, 34-5
  - Method, 34-2
  - Other Fields, 34-5
  - Parameters, 34-4
  - Pause, 34-3
  - Play, 34-2
  - Play Segment, 34-3
    - Parameters, 34-4
  - Record, 34-3
  - Rewind, 34-2
  - Search, 34-2
    - Parameters, 34-4
  - SearchBk, 34-3
    - Parameters, 34-4
  - Stop, 34-2
  - Type, 34-2
  - Video Editing, 34-6
  - Wait For, 34-5
  - Zero\_HMS, 34-4
- VCR Port, Appendix H-8
- VCR Type (Sony VISCA), Appendix H-7
- Vdisc Event, 5-8, 36-2
  - Audio1, 36-3
    - Parameters, 36-3
  - Audio2, 36-3
    - Parameters, 36-3
  - Chapter Example, 36-4
  - FrameDis, 36-2
    - Parameters, 36-3
  - GetFrame, 36-2
    - Parameters, 36-3
  - Method, 36-1
  - Parameters, 36-3
  - PlayBKW, 36-2
  - PlayFWD, 36-2
  - Playseg, 36-2
    - Parameters, 36-3
  - PlaysegP, 36-2
    - Parameters, 36-3
  - Search, 36-2
    - Parameters, 36-3
  - Sendcom, 36-3
    - Parameters, 36-4
  - StepBKW, 36-3
  - StepFWD, 36-3
  - Stop, 36-2
  - Type, 36-1
- Version, PLComm, 25-3
- VDISCAUD, Environmental Variable, Appendix H-16
- VidBW, Video, 35-18
- Video, Black & White, VidBW, 35-18
- Video Brightness, Hue, Contrast, Saturation, Appendix H-6
- Video Display section, 35-2
- Video Editing, VCR, 34-6
- Video Event, 5-7, 35-1
  - AAPlay, 35-10
  - Animation section, 35-9

- CEGMODE, 35-7
- Chapter Example, 35-2, 35-21
- ClockOff, 35-8
- ClockOn, 35-8
- CycleCol, 35-9
- FadeIN, 35-4
- FadeOUT, 35-6
- Infowindow section, 35-19
  - Infowind, 35-19
- Introduction, 35-1
- LoadPal, 35-2
- Motion Video section, 35-15
- ResetPal, 35-3
- SavePic, 35-4
- Scrollup, 35-11
- SetMode, 35-3
- Ticker, 35-13
- VidBW, 35-18
- Video Display section, 35-2
- VidGrab, 35-17
- VidInv, 35-18
- VidLive, 35-15
- VidNorm, 35-16
- VidOver, 35-15
- VidTcol, 35-17
- VidWind, 35-16
- ViewOFF, 35-7
- ViewPort, 35-6, Appendix H-21
- Video Input Standard (NTSC/PAL), Appendix H-6
- Video, Inverse, VidInv, 35-18
- Videodisc Serial Port Parameters, Appendix H-5
- VidBW, Video, 35-18
- VidGrab, Video, 35-17
- VidInv, Video, 35-18
- VidLive, Video, 35-15
- VidNorm, Video, 35-16
- VidOver, Video, 35-15
- VidTcol, Video, 35-17
- VidWind, Video, 35-16
- View Directory, 2-4
- ViewOFF, Video, 35-7
- ViewPort, Video, 35-6, Appendix H-21
- VMP\_FORMAT, Environmental Variable, Appendix H-16
- VOC Files, 6-1
- VVDAC, Environmental Variable, Appendix H-17, H-20

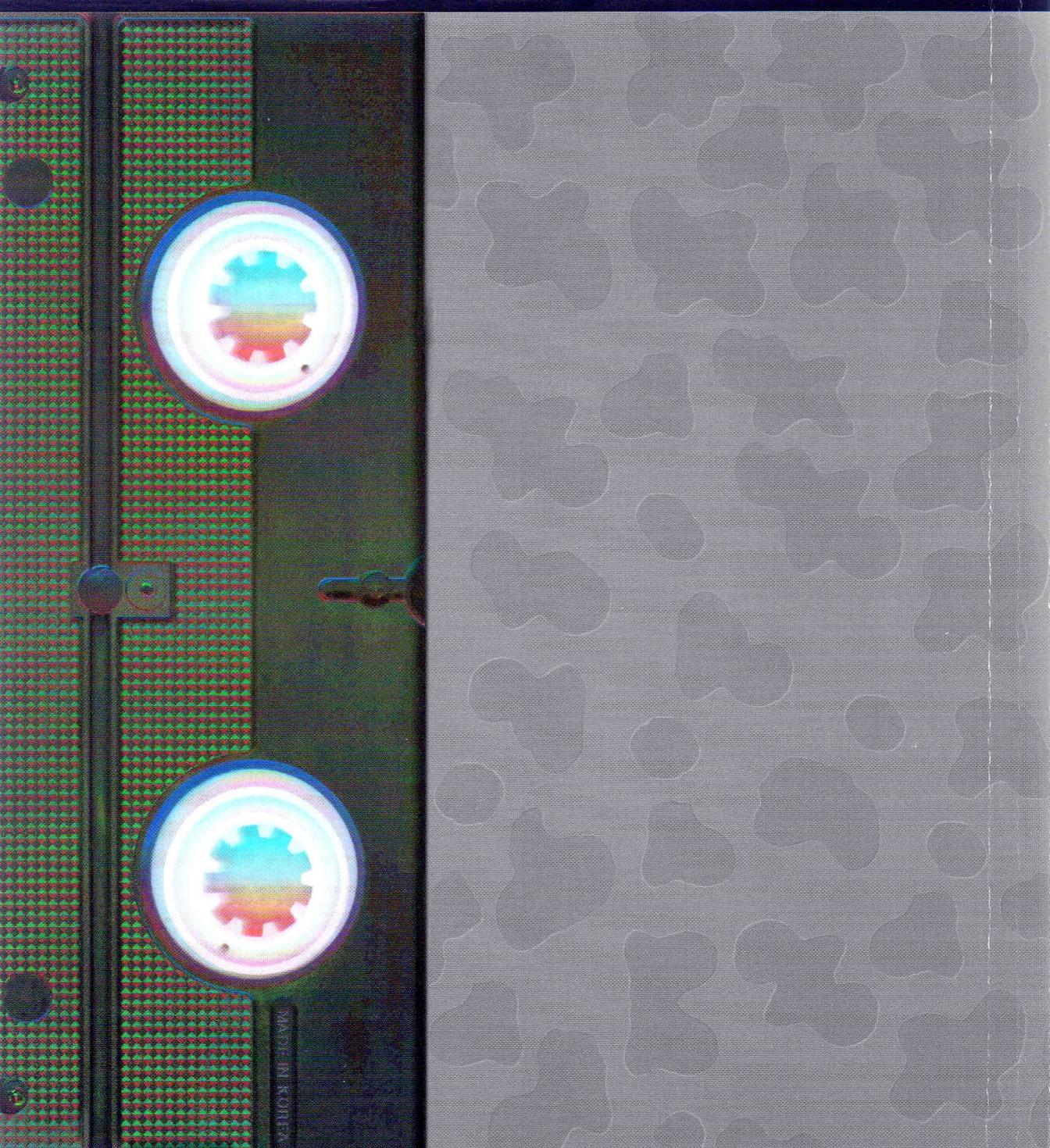
- VVLINES Environmental Variable, Appendix H-16, H-18
- VVMODE Environmental Variable, Appendix H-16, H-19
- VVPAINT Environmental Variable, Appendix H-17
- VVPIC Environmental Variable, Appendix H-17
- VVSHARE Environmental Variable, Appendix H-17
- VVSPRITE\_BUF Environmental Variable, Appendix H-18
- VV\_VARK Environmental Variable, Appendix H-18

## W

- Wait Field, 2-11
- Wait For Any Key, 4-3
- Wait For Field, 4-3
- Wait For Timeout, 4-3
- WAV Files, 6-1
- Width/Height, Text, 29-4
- WIN Files, 19-4, 27-1
- Windows 3.0/3.1, 1-11
- Wipe Menu List, 7-2
- WND Files, 18-1
- Word Button, 2-6
- Work Button, 2-6
- Workform, 3-1
- Write
  - DBMS, 11-7, 11-15
  - Var, 33-3, 33-12
- WriteFld, DBMS, 11-7, 11-14

## Z

- Zero\_HMS, VCR, 34-4



mathematica™  
INCORPORATED

