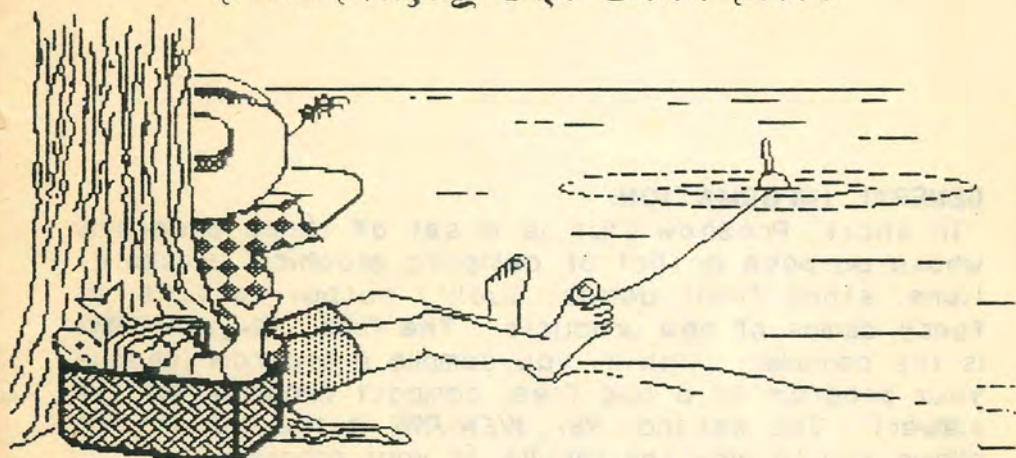


# ProShow Plus

"With ProShow Plus You Can Relax!"  
"And Enjoy The Good Life"



## ProShow Plus

*Professional Slide Show  
by Dwayne A. Hickman  
of StarSoft Development Laboratories  
Version 1.0*

Atari ST

Atari ST

Atari ST



**1988 STARSOFT DEVELOPMENT  
LABORATORIES**

# ProShow Plus

*Professional Slide Show Instructions*  
*by Dwayne A. Hickman*  
*of StarSoft Development Laboratories*  
*Version 1.0*

## **GENERAL INFORMATION:**

In short, ProShow Plus is a set of three programs whose purpose is that of designing graphical presentations, store front demos, quality output to VCRs, or fancy demos of new products. The first, *PS+\_COMP.PRG*, is the compiler. With it, you compile a program (change your program to a bug free, compact program for the viewer). The second, *PS+\_VIEW.PRG*, is the viewer that allows you to view the results of your program.



The following text describes some aspects of the four programs that make up ProShow Plus. Please read. All ProShow Plus programs are designed to be run in medium resolution. Both the compiler (PS+\_COMP) and the animator (MAKE\_ADP) may only be run from the SHELL environment.

**SHELL:** (the shell)

This program must be run in medium resolution. It verifies authorized users and allows the user to go between the programs that make up ProShow Plus without returning to the desktop. You may select the program you wish to run by pressing the number or letter corresponding to it (on the left hand side), or by placing the mouse over the name of the program and clicking either button.

**COMPILER:** (the compiler)

The compiler allows the use of accessories. Please be aware of some unfriendly ACCs.

The Touch-Up editor is not really designed for writing programs although it may be used for that purpose.

Read HELP screen under OPTIONS menu for details on how to use the compiler effectively.

You may chain directly to PS+\_VIEW.PRG from the compiler; don't have to go back to the SHELL. Just select PS+\_VIEW from the menu.

**VIEWER:** (the viewer)

- 1) Run or double click PS+\_VIEW.PRG
- 2) READ, then close credit box.
- 3) Load in a file. Must end in .PSD and be a correct version--Now 1.0

- 4) Click close Complementary Disk Change box--this box allows you to have your pictures on a different disk than the disk with the ProShow Plus programs.
- 5) The program is now running. Earthquakes, nuclear wars excepted.
- 6) Follow the prompts given by the program.
- 7) You may press HELP key to view any screen, or press ESC key to freeze the program--unfreeze by pressing the SPACE BAR.
- 8) You may draw on the screen with the left/right mouse button--depending on the program--not active all the time.
- 9) Click closed the EXIT button to return to the DESKTOP, or AGAIN to view the show again.

If the viewer did not load in all of your pictures, then check to see if there was enough memory. You may have to (should) remove ACCs, RAM resident programs, and RAM drives. If not this, then is the number in MAX\_SCRN.DAT too small? ProShow Plus may hold up to ten (10) screens in memory at one time on a 512K system. A maximum of 60 screens is ProShow's Plus's current limit. Screen zero (0) denotes the visible screen.

NOTE: the last screen is used as a buffer for decompressing pictures (\*.CAP); take heed when using this screen along with CAP pictures.

**MAKE\_ADG:** (the animator)

Read the HELP screen after running MAKE\_ADG.PRG for details on how to effectively use this program. This



program allows the user to easily create complex color cycling pictures with multiple palettes. MAKE\_ADP will load and save the following picture formats: NEOchrome, Degas (PI1 and PI2), ADP, and CAP.

**Other files:**

**MAX\_SCRN.DAT**

Contains a number from 1-60 representing the maximum number of screens you want ProShow Plus to contain that run. If this file is not found, then ProShow Plus will consume every bit of memory it can get. Specifically, the amount of memory will be:

Memory-Minimum ( max\_scrn , (Free memory-20000) div 32256 ]\*32256 )

One reason not to consume all memory is if you want to use the EXEC command or if other programs in memory may request more memory.

NOTE: If the number in MAX\_SCRN.DAT is positive then ProShow Plus will try to reserve that number of screens. If it fails it will continue after reserving all it could. If the number is negative, then ProShow Plus will exit back to the shell if it failed in reserving the requested number of screens.

**PS\_ML.BIN:**

If this file is not on the same disk as PS+\_VIEW.PRG then you will not go anywhere. This file contains the machine language routines that ProShow Plus uses for fast picture decompression, for a couple CLEARs, and for a few BLENDS. It is crucial for PS+\_VIEW.PRG's running. It is also required to run MAKE\_ADP.PRG.

**Distribution notes:**

When you wish to release a PS+ program to Public Domain, to other people, or include one on a commercial disk, then note the following:

- 1) PS+\_VIEW.PRG is free to be released into Public Domain. This program must be on the disk you plan to distribute.
- 2) PS\_ML.BIN must also be on that disk.
- 3) Your ??????????.PSD compiled file must be on that disk.
- 4) Your picture files (NEO/PI?/ADP/CAP) must be on that disk.
- 5) Your source code for the PS+ program (the one compiled to ??????????.PSD) MAY be on that disk at your option.

For example, a directory of a disk that contains a PS+ program to show off three pictures may be:

PS+_VIEW.PRG	35K	PS_ML.BIN	1K
SHOW_OFF.PSD	1K	PIC_1.NEO	32K
PIC_2.ADP	32K	PIC_3.CAP	7K.

**ATTENTION:**

All other programs on the original ProShow Plus disk are NOT to be distributed in any way. They are copyrighted material for the consumer's use only. Distributing, selling, posting them on a BBS, or in any way allowing illegal access to them (and the manual) will incur a violation of current copyright laws. Please respect the programmer's work any obey the law.



### **COPY PROTECTION REMOVAL:**

If you find yourself using ProShow Plus often, and are becoming annoyed with the copy protection, then you may purchase a non copy-protected package for \$29.95. This package is identical to the original except:

- 1) There is no copy protection
- 2) Each disk is tagged with your company's identifying information.
- 3) Copy of PS+ Companion disk
- 4) Special notification of upgrades

You must present a valid mailing address and a working phone number (we will call you) in order to buy the non copy-protected version. You must also sign and return a short form saying you understand the copyright laws and their corresponding penalties. The ID tagging allows us to uniquely identify any copies illegally placed in public domain or on any bulletin boards.

**Mail requests to the author:**

c/ Dwayne A. Hickman, ProShow Plus  
StarSoft Development Laboratories  
331 Bartlett Blvd.  
Orlando, Florida 32811

**COMING SOON:**

While ProShow Plus is a powerful language, not all methods of achieving excellent results may be obvious to the user. Hence StarSoft will distribute a companion disk to ProShow Plus. This disk will contain many helpful hints for doing impressive graphics from within PS+ and many utility programs to help you easily design special pictures (NEO, Degas, and ADP) that can be used by PS+ to give eye catching presentations. Order the ProShow Plus companion disk from TDC Distributors for only \$12.95.



---

**ProShow Plus**  
**Professional Slide Show Instructions**  
**by Dwayne A. Hickman**  
**of StarSoft Development Laboratories**  
**Version 1.0**

---

**GENERAL INFORMATION:**

In short, ProShow Plus is a set of three programs whose purpose is that of designing graphical presentations, store front demos, quality output to VCRs, or fancy demos of new products. The first, *PS\*\_COMP.PRG*, is the compiler. With it, you compile a program (change your program to a bug free, compact program for the viewer). The second, *PS\*\_VIEW.PRG*, is the viewer that allows you to view the results of your program.

Therefore, the algorithm for designing a slide presentation is:

- 1) write the program (use any editor/word processor you like but save the file in ASCII format--most all editors/WP allow for this)
- 2) run *PS\*\_COMP.PRG* and compile your program. Since many programs will have a few bugs in them, the compiler also has a simple editor built into it to fix your typos quickly.
- 3) run *PS\*\_VIEW.PRG* and view your results. If they are not exactly what you wanted, then repeat the above step(s).

And the last, *MAKE\_ADP.PRG*, is a program to design animation (more technically known as color cycling--but called here, and throughout this document, animation) by

the use of up to seven different palettes. This program will load *ADP*, *CAP*, *NEOchrome*, and *Degas* files and also it will save pictures in any of these formats. Both the compiler and the animator have help screens available in case you need some quick help/information.

### THE LANGUAGE:

The current ProShow Plus (PS+) language consist of 52 commands. Below is a formal description of each, and why you might use it in your program. For brevity, let 'n' and 'm' denote numbers. The commands are primarily listed by syntax and secondarily sorted alphabetically.

-----

\* <-- This is a single quote. Anything may come after it.

**Remark.** Allows the programmer to describe what their program is doing. This statement is ignored during compilation.

**NOTE:** when errors are detected during compilation, the corresponding statement is converted to a remark. [i.e. since [BLEND 3,Junk] is a bad statement, it will be converted to [\*\*\*ERR->BLEND 3,Junk]

### ANIMATE

This is the initial setting. This command tells the viewer to follow the animation status of the picture when displaying a picture by a BLEND command.

### ANIMATE OFF

Will NOT animate any picture displayed by a BLEND, irregardless of the picture's animation status. In general, these three commands are only used while debugging a program or designing special effects.

### ANIMATE ON

Will animate EVERY picture displayed by a BLEND, irregardless of the picture's animation status.



## **BELL**

Makes a bell sound to alert the user to notice something.

## **DEMO OFF**

This is the default. All options work normally. Use this for presentations or for user active demos.

## **DEMO ON**

Tells the viewer (*PS- VIEW*) not to allow the user to have the ability to press keys to: enter HELP, bypass scrolling, bypass animation, or exit out of loops. WAITKEY is left unchanged.

This command is primarily for those programs that are going to be shown as demonstrations and user input is mostly ignored--else they could exit the program, and we wouldn't want that, would we? Also, no on screen drawing with mouse even if DRAW is set to ON.

## **DISK**

Prompts user to change disk in the current drive. This command is primarily for single disk drive users. The normal GEM prompt to change disks may be messed up if the screen's resolution is not set to medium. Clears *Main Screen* and sets resolution to medium.

## **DRAW OFF**

Turns off the mouse's ability to draw.

## **DRAW ON**

Allows the user to draw on the screen with the mouse buttons. The left button draw with the color selected by COLOR. The right mouse button draws with COLOR+1.

Drawing may be done only during a WAIT or a WAITKEY] with an active (visible) mouse and DEMO OFF. Useful for highlighting screen areas during a presentation.



## **END**

Exit program and ask user if they wish to rerun the current program. This alert/input is with an alert box. Clears *Main Screen* and sets resolution to medium.

## **FADE**

Fades the colors on the screen to black. This is useful if the pictures you are switching between have different resolutions or very different palettes. Ignored if previous FADE not restored.

## **HELP**

Forces the program to jump to HELP SCREEN--also can be jumped to by the user during a WAIT/ WAITKEY while the program is running. The HELP SCREEN shows what pictures are in what locations. Therefore, if you need to reshow a picture after your program has gone past it, press HELP and choose which picture to display irregardless of what picture your program said to display.

Displays during HELP are of BLEND n,1 type with NO animation.

Not available if DEMO is set to ON.  
Clears *Main Screen* (screen 0) and sets resolution to medium.

## **LIGHT**

Restores a FADE. Ignored if no FADE.

## **REPEAT**

Usually placed at the end of your program. Will start rerunning your program from the beginning. Useful if you want the program to run indefinitely--i.e. over night in your store front window.

No prompt for rerunning will be given.

Hence, if you want the program to run with no user input then do NOT use this command with a START command.

## **START**

Makes a GEM alert box. Since loading in a lot of picture files may take some time, this allows you to stop once you load them in and wait till the user clicks the box closed before going on with the show.

NOTE: if no LOAD commands are performed after a START command, then when ask to rerun your slide program, you will not have to wait to reload the picture files the second run. If a LOAD command comes after a START command, then PS- will assume that the load destroyed previous picture(s) and hence will reload all pictures like it did the first run. If any LOAD loads a picture in a location that already contains a picture, then PS- will reload all pictures if the program is rerun regardless of the START rule.

In general, you will use a START command after LOADS to tell the computer to wait for your signal to continue--the alert box will prompt you with a START button. Close it and then the program will continue.

Clears *Main Screen* (screen 0) and sets resolution to medium.

## **QUIT**

Exits PS-*\_VIEW* and returns to Desktop. Colors and memory are restored before returning to the Desktop.

.....

## **CLEAR n**

Clears a screen to the background color (color 0). The number n specifies which method to use to clear. For those of you with hardware that allows you to overlap computer images and broadcast images, the CLEAR command (with bleed through on) will actually allow the



broadcast image to show though the screen parts cleared. Hence, this is like a BLEND command for this special situation. The bleed through option is selected by your hardware not by PS-.

CLEAR is very useful if you are changing pictures with different resolutions, or pictures with very different palettes.

n-(1..11)

#### **COLOR n**

Allows the program to specify which color number to do the DRAW with, or the color number of the text during a TYPE/TEXT command.

n-(0..3)     \*\*\* Soon to be n-(0..15) GFA-BASIC problem.

#### **MOUSE n**

Allows program to switch shape/visibility of the mouse to GEM standards.

- |                    |                             |
|--------------------|-----------------------------|
| 0-Arrow            | 1-vertical bar              |
| 2-bee              | 3-hand with pointing finger |
| 4-flat hand        | 5-thin cross-hair           |
| 6-thick cross-hair | 7-outlined cross-hair       |
| 8-not visible      | 9-visible.                  |

n-(0..9)

#### **PAUSE n**

Waits n/100th of a second. Useful for small delays between SOUND, PALETTE, SETCOLOR, and between BLOCK commands. For larger delays use the WAIT command.

n-(1..100)

#### **REZ n** (rez is short for Screen Resolution)

Allows the programmer to force a change in resolution. Where 0 is low rez, and 1 is medium rez. Since BLENDS change rez after merging a picture, you may wish to change it before calling a blend. A change in rez will unavoidable clear the screen; however a REZ will have no effect if n equals the current resolution.

n-(0..1)



**SOUND n**

Plays the preset sound effect n. Use with PAUSE in loops for more elaborate sound effects.

1-smacks    2-explosion    3-bounce    4-cords  
5-saucer    6-vibro    7-tones    8-bells  
9-buzz

n-(1..9)

**SPEED n**

Determines the speed in which most of the BLENDS will operate at. The normal speed is n-1. Setting n-0 should be avoided as it will probably yield flickering during some BLENDS. Used for special effects such as slowly opening a stage curtain.

The larger the number n is, the longer it takes for the BLEND command to complete.

n-(0..60)

**WAIT n**

Waits n seconds. HELP may be called during this wait. Drawing is also possible.

n-(1..999)

**WAIT n FOR KEY**

Waits n seconds or for any key to be pressed. Which ever comes first.

n-(1..600)

**WAITKEY n**

Waits until a key with ASC code equal to n to be pressed.

HELP may be called during this wait.

Drawing is possible.

ASCII tables may be found in many PC sources. "A" is 65, "a" is 97, " " is 32, etc.

n-(1..255)

---

**LOOP n**

Executes all statements between corresponding LOOP and LEND commands n number of times. You may have up to five imbedded loops. The compiler automatically indents to line up commands in a loop.

n-(0..999)

**LEND**

Ends a loop.

**EXIT ON n**

Exits the current loop if a key with ASCII code equal to n is pressed.

n-(1..255)

---

**BLEND n,m**

Blends picture n onto the screen by the method m--this number's meaning is found on the SHEET\_of\_BLENDS.

BLEND is the main power command behind ProShow Plus. When using PS- you should have the SHEET\_of\_BLENDS and SHEET\_of\_BLOCKS/CLEARs handy. Experiment with the different BLENDS to get used to what they actual look like on the ST.

If you have any good ideas for new BLENDS, send them to us and we shall try to incorporate them in future updates of ProShow Plus. Please keep in mind the power/limitations of the Atari ST line of computers. If possible, send source code.

Avoid blending pictures with different resolutions (use CLEAR/REZ and then call a BLEND command).

Note that on the SHEET\_of\_BLENDS, some BLENDS are specifically designed for use in low resolution only.

NOTE: the algorithm that most BLENDS follow is that they first merge the screens, update the color palette, and then the BLENDS correct the screen's resolution, if that's necessary. If you want to correct the rez or you



want to change to the new screen's palette first, then use the REZ or the PALETTE commands.

n-(0..60) m-(1..27)

### **BLOCK n,m**

Gets a block starting at the upper left corner (WIDTH-n, HEIGHT-n) and lower right corner (WIDTH-m, HEIGHT-m). Next, it proceeds to place the block at the position starting at (PLACE-n, PLACE-m). If it extends off the screen then the results may be less than predictable. n specifies which picture location to get the block from, and m specifies which method of showing the block to use.

n-(0..60) m-(1..6)

### **COPY n,m**

Copy screen-n to screen-m (including palettes, etc). Useful with OVERLAY command. Unless a picture was COPIED to (or LOADED to) the main screen (SCREEN 0) do not copy the main screen to another location.

n-(0..60) m-(0..60)

### **FILM n,m**

Show an animation sequence of the picture in location m. FILM shows a sequence of 16 frames. n specifies the delay between frames--1/60th of a second for every increment of n. The frames are taken from picture m using the following rule (LOW REZ RULE).

- i) go left to right, from top to bottom
- ii) frames are 80 pixels wide by 50 pixels high
- iii) frame-1 is (0,0)-(79,49)  
frame-2 is (80,0)-(159,49)
- iv) frame-5 is (0,50)-(79,99)  
frame-6 is (80,50)-(159,99) ...

The frames are flipped through (put at) the location specified by the PLACE command.

NOTE: because of flickering, PS- version 1.0 will mod the x-coordinate to a multiple of 16 pixels in low rez or to a multiple of 32 pixels in medium rez.

n-(0..60) m-(0..60)



**HEIGHT n,m**

Sets height for BLOCK command.

NOTE: Blocks can only be up to 1/4 the size of a full screen.

n-(0..199) m-(n..199)

**OVERLAY n,m**

Overlay screen-n on top of screen-m. If used properly, this is a powerful command for obtaining complex picture transitions. Also very useful with best picture compression. I.e: complex background (large \*.CAP) with 20 overlays (small \*.CAPs).

NOTE: Overlay is designed for LOW REZ ONLY.

n-(0..60) m-(0..60)

**PALETTE n,m**

Will change the current palette to that specified by the m-th palette of the picture in location n. This command is used for program controlled animation. All pictures have seven (7) palettes--even *NEO* and *DEGAS* pictures have seven palettes, though all seven are the same.

n-(1..60) m-(1..7)

**PLACE n,m**

Sets (x,y) coordinate to start putting the block at on a BLOCK command. Also sets the start position of where the text from the TEXT or TYPE commands will begin to print at.

n-(0..639) m-(0..199)

**ROTATE n,m**

Will rotate the lower 50 scans lines at a speed of n for m times. There is no flickering although if you select the slow setting then the halt/start of the movement may make any text in the moving block annoying to read.

n-1 is fast, n-2 is medium, n-3 is slow.

n-(1..3) m-(1..999)

### **SETCOLOR n,m**

Allows program to change color number n to represent the color m. Here 'm' represents the RED|GREEN|BLUE intensities.

Intensities are between 0 and 7 inclusive. White is 777. Black is 000. Red is 700. This command along with the PAUSE command are useful for simple animation (color cycling).

See PALETTE/ *MAKE\_ADP.PRG* for more complex animation.  
n-(0..15) m-(0..7|0..7|0..7)

### **WIDTH n,m**

Sets width for BLOCK command.

NOTE: Blocks that are on non-hex boundaries take longer and they tend to flicker if PS+ is run without the blitter set to on. Keep this in mind when drawing your pictures if you intend to use the block functions for animation purposes or if you use BLOCKs 5 or 6.

NOTE: Blocks can only be up to 1/4 the size of a full screen.

n-(0..639) m-(n..639)

### **WORD n,m**

Will scroll upward at one scan line per 60th of a second (fastest possible for the smoothest movement) the nth bit plane. The top will wrap around to the bottom. This is designed for low resolution although it does have interesting effects on medium resolution screens. m represents the number of times to scroll upward.

WORD 1,800 will scroll the entire screen 4 times (200\* 4=800) without any pauses--very smooth.

NOTE: A COPY command that copies *Main Screen* to a screen in memory should NOT be issued after using this command. I.e. COPY 0,3 should be avoided after WORD.

n-(1..4) m-(1..999)



-----

**EXEC n,file\$**

Stops running *PS+\_VIEW.PRG* and starts running *file\$*. The number *n* is used to reserve enough memory to allow the execution of the other program. If that program crashes, then *PS+* probably will also.

NOTE: Not all programs are executable by EXEC.

Beware, this command is not BOMB proof. In fact, this is the only known command/way of crashing ProShow Plus under normal operations.

Upon a successful completion of the other program, *PS+\_VIEW* will resume running. Useful for adding music to ProShow Plus viewer by running background music programs. Also you can run small programs that draw pictures on the screen.

*n*-(1..999)

**LOAD n,file\$**

Loads a picture file *file\$* into the picture location *n*. This number *n* is the number to use with the BLEND, BLOCK and PALETTE commands. *PS+* can load *ADP*, *CAP*, *NEO*, *PI1*, and *PI2* pictures. *file\$* is any valid pathway and/or file name. I.e.:

LOAD 3,C:\PICTURES.PI7\PICTURES.PI1\MAN.PI1

is valid. Read the command START's function based on loads.

NOTE: the compiler does not verify correct pathways/ file names.

NOTE: the last 12 character of *file\$* will be shown on the HELP SCREEN to report which picture was loaded (if) where. *CAP* pictures are buffered in the highest screen. YOU ARE WARNED. For example, if there are eight screens available, *CAP* files will be loaded into screen-8, decompressed, and then moved to the assigned screen number. Whatever was in screen-8 will be overwritten with the compressed data. This gives the programmer more memory to play with as *PS+* does not need a



special buffer for decompressing *CAP* pictures.

Vertically XORed *CAP* files are decompressed in about 1/5th of a second. Horizontal ones in about three (3) seconds. *CAP* files are (C)ompressed (A)DP (P)ictures. Use *MAKE\_ADP* to compress *ADP*, *NEO*, *PI1*, or *PI2* pictures to *CAP* files. *CAP* files average 9Kb-10Kb.

n-(1..60)

### **SCROLL n,text\$**

Will scroll (right to left) the text *text\$* across the bottom of the screen until the last character of *text\$* has exited off the screen. The number *n* specifies the speed the text *text\$* will be scrolled at. If a key is pressed during the scrolling, then the scrolling will end and the program will continue. The bottom nine lines of the screen will be cleared before the text is scrolled and then replaced afterwards. The color of the text will be color 15 (Low rez) or color 3 (Medium rez).

n-1 is fast, n-2 is medium, n-3 is slow.

n-(1..3)

### **TEXT n,text\$**

Places text *text\$* of size *n* at location starting at coordinates specified by the *PLACE* command. Text color is that specified by the *COLOR* command.

n-(1..40)

### **TYPE n,text\$**

Starts typing *text\$* at location specified by *PLACE* command at a speed of *n*. n-100 is one second. Text color is that specified by *COLOR*. Text size is the standard medium/low rez text size.

n-(1..100)

-----

**GOSUB :??????**

Unconditionally branches to some other part of the current program. If the label is not found, then the GOSUB will be ignored. Useful for having many places in your program call the same code (lines of commands) without having to retype the same sequences of commands for every use.

When a RETURN is detected, then the command directly after the last calling GOSUB will be executed.

Up to ten (10) levels of nested GOSUBs may be used. If an eleventh is called, then it will be perform but the tenth will be forgotten about.

**GOTO :??????**

Unconditionally branches to some other part of the current program. If the label is not found, then the GOTO is ignored. Primary for debugging to skip over debugged code. Will be more useful in future ProShow Plus versions that support conditional branching. It may be used to create infinite (never ending) loops

**NOTE:**

The syntax ":?????????" represents a label that is to be defined by the LBL command. The colon (:) must be present. Any typable character may be substituted in place of the question marks (?).

When the compiler reads and flags labels, it recognizes capital and lower case letters as distinct. Therefore, the label ":ABC" is completely different than the label ":abc" as far as the compiler is concerned.

Labels have a minimum of zero characters (i.e. ":") and a maximum of eight characters (i.e. ":12345678").

While blanks are recognized if they are between other characters (":HI THERE"), they are not recognized if they are the ending characters in the label (":Hi " is bad.).



### **LBL :??????**

Allows user to specify labels in their ProShow Plus program. GOSUBs and GOTOS use these labels for branching program operations. ? denotes positions where characters may be substituted. Any typeable character may be used. [i.e. Valid labels:

:AAA	:AAAAA	:()67()	:.<.>Hi
:.....	: hi hi	:*<>[]::	:Z

Labels may have up to six (6) characters identifying them. Labels may be repeated but only the first occurrence will be branched to.

### **ON KEY GOSUB :??????**

Same as GOSUB but will be executed when a key has been pressed not when the command is parsed. For example, if you put "On key Gosub :A" at the beginning of your program, then PS- will parse and remember it. PS- will then go on its merry little way and execute the next commands. Now, when the homely little kid walks by and presses any non-null key on the terminal, PS- will jump to label :A and do whatever the commands after :A say to do.

Allows the programmer to set up their demo programs so that the demos will run continuously if no one is messing with the ST and to jump to the beginning of the demo if someone presses a key.

### **ON KEY GOTO :??????**

Same as ON KEY GOSUB but follows rules of GOTO instead of GOSUB.

### **RETURN**

Terminates a subroutine (called by GOSUB). Your program will continue with the next statement after the calling GOSUB. Up to ten nested subroutines may be made. If ten GOSUBS are active and an eleventh GOSUB is called, then the tenth will be replaced with the eleventh. The tenth will be forgotten.

-----

**SPECIAL NOTES:**

At start up, the *PS+\_VIEW.PRG* looks for a file named *MAX\_SCRN.DAT*. This file contains a number that the user may specify that sets the maximum number of screens used by the viewer. A negative sign in front of the number tells the viewer to stop running if that many screens were unavailable. A positive number tells the viewer to continue even if the viewer did not receive the requested number of screens. If the file was not found, or there was garbage in it, or the number was too large for the memory configuration, then the actual amount of memory consumed would be:

dummy = maximum ( [Free Memory - 20K] Div 32K , max\_scrn.dat )  
No. of screens = minimum ( dummy , 60 )  
memory consumed = ( # of screens \* 32256 + 20K ( +38K for PG+ )

The 20K is needed for BLOCKS, Alert boxes, and general variable storage. This allows up to ten screens to be held in an unexpanded 520ST. And up to 60 in a 4-Meg system. Screen 0 (the actual viewing screen) may also be used to get up to eleven (11) screens on a 512K 520ST--TOS in ROM. If you attempt to load files into a location and there is not enough room for that location, (i.e., have 1-7 available and do a LOAD 8,PIC.NEO) then that file will NOT be loaded and the picture for that number will be set to picture location one (1). If a bad file/pathway, or the file was not found, that command will be skipped and a flag will be displayed on the HELP screen saying the file was not loaded.

To EXIT *PS+\_VIEW.PRG* once it is running, press HELP key at a wait, and enter -1 at prompt. Then click EXIT button to return to desktop. This is not available if DEMO is set to ON.



---

## FREEZE

Press the Esc key to freeze the current program. If program is frozen, press SPACE BAR to unfreeze the program.

This feature is not available if the program is set to DEMO ON.

---

## COMMENTS:

*GFA-BASIC* programs tend to render GEM windows sluggish after program termination. Everything works but may take a couple tries the first time to resize a window. After the initial sluggishness, everything returns to normal.

---

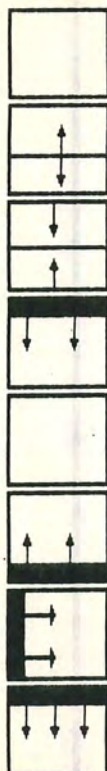
## SHELL

Only the program *PS+\_VIEW.PRG* will run without the *SHELL* program first being run. You must run *SHELL.PRG* first to run *MAKE\_ADP* and *PS+\_COMP* programs.

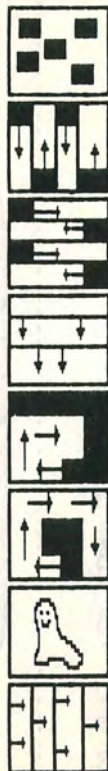
---



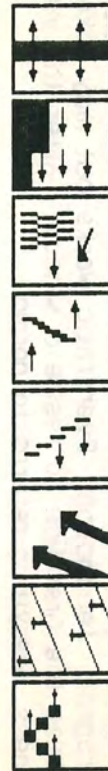
The End



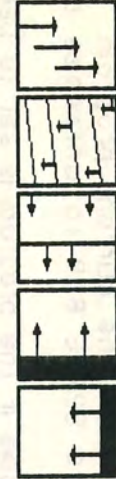
- 1. INSTANT
- 2. UNROLL
- 3. UNROLL
- 4. UNROLL
- 5. FADE INSTANT LIGHT
- 6. SCROLLUP
- 7. UNROLL
- 8. SCROLL DOWN



- 9. RANDOM BLOCKS
- 10. UNROLL
- 11. UNROLL
- 12. MERGE
- 13. MAZE IN
- 14. MAZE OUT
- 15. GHOST
- 16. MERGE



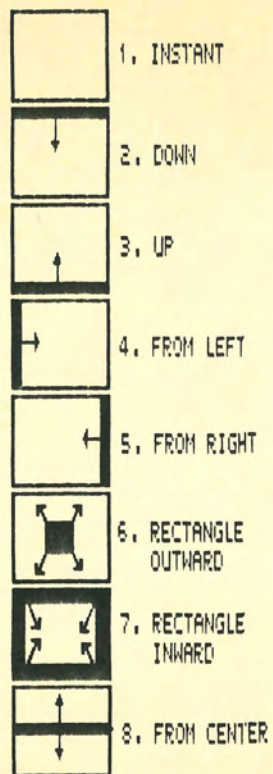
- 17. SCROLL OUT
- 18. BLOCKS LEFT
- 19. SMEAR
- 20. SMEAR
- 21. SMEAR
- 22. SMEAR
- 23. SMEAR
- 24. SMEAR



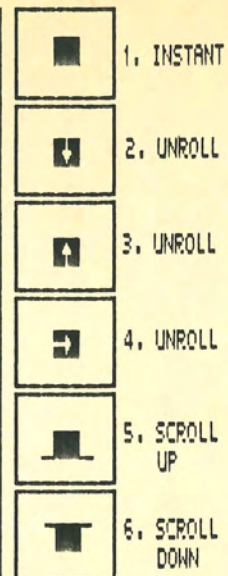
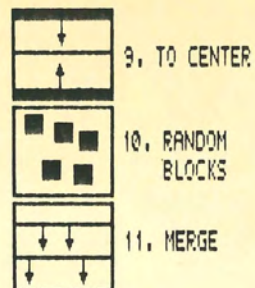
- 25. SMEAR
- 26. SMEAR
- 27. MERGE BIG STEPS
- 28. UNROLL
- 29. UNROLL

BLEND 1-29 (C) 1988 BY StarSoft Development Laboratories





CLEAR 1-11



BLOCKS 1-6





Card: 02

X	1	2	3	4	5	6	7	8	9	10
01	934	603	273	946	623	302	984	667	352	038
02	382	301	222	149	081	016	955	897	840	786
03	846	029	218	414	617	824	036	251	469	690
04	320	778	245	720	203	692	187	686	189	694
05	801	541	291	053	823	602	386	176	970	767
06	286	313	353	405	468	540	619	705	795	889
07	776	093	425	772	131	500	878	262	652	047
08	268	878	505	149	807	477	155	841	534	231
09	762	667	592	535	494	466	447	437	434	437
10	257	459	684	928	190	465	751	047	350	660
11	754	254	779	326	892	472	064	667	278	896
12	253	052	878	729	599	486	385	296	216	143
13	752	851	980	135	311	505	713	932	162	400
14	251	652	084	544	027	529	045	575	115	663
15	752	454	190	956	746	556	383	222	073	934
16	253	257	292	370	468	587	724	874	037	209
17	754	061	407	786	192	621	068	530	005	490
18	256	866	517	203	919	657	415	189	976	775
19	759	671	628	622	647	696	765	851	951	063
20	261	478	741	043	376	736	117	515	928	354





Card: 04

-----

X	1	2	3	4	5	6	7	8	9	10
01	362	259	160	066	976	891	809	730	653	577
02	809	756	711	675	646	625	608	596	587	582
03	287	315	356	409	473	546	626	713	804	899
04	786	915	061	224	400	588	785	990	201	417
05	299	542	807	093	396	712	040	377	721	072
06	821	188	583	002	440	895	363	841	329	825
07	350	849	380	939	520	120	736	364	002	649
08	885	521	193	898	627	378	147	929	724	528
09	424	201	020	873	756	662	588	529	483	449
10	966	888	856	863	901	966	052	155	273	404
11	511	580	700	862	060	286	535	803	087	385
12	058	276	551	871	229	619	033	468	921	389
13	607	977	407	887	408	963	544	148	771	410
14	158	680	268	909	595	316	066	840	635	447
15	710	386	132	937	787	676	597	543	511	497
16	263	094	000	968	986	044	135	254	396	558
17	817	804	871	004	189	417	680	973	290	628
18	371	515	745	042	396	795	231	698	191	706
19	927	229	620	084	607	177	787	429	099	792
20	483	943	497	128	820	563	347	165	012	883

Card: 05

X	1	2	3	4	5	6	7	8	9	10
01	606	649	696	750	809	872	940	010	083	158
02	084	108	142	186	241	302	370	444	521	602
03	601	645	704	780	868	967	075	190	311	437
04	144	232	343	473	621	782	955	137	327	523
05	703	853	032	236	460	701	955	221	497	780
06	274	499	758	046	359	692	041	403	776	159
07	855	163	511	893	303	737	189	656	137	628
08	442	840	284	767	282	823	385	964	559	165
09	035	528	073	662	287	940	617	314	028	755
10	631	224	875	574	312	083	879	698	534	386
11	232	927	687	499	355	246	165	108	071	051
12	834	635	507	436	411	425	470	541	633	744
13	440	348	334	381	479	618	791	991	215	459
14	047	065	166	334	556	822	125	457	815	194
15	655	785	003	293	641	036	470	936	429	944
16	265	508	845	258	733	258	825	426	055	709
17	876	233	690	228	831	488	188	925	692	485
18	489	960	538	202	934	723	559	432	338	271
19	102	689	389	179	042	964	935	947	992	066
20	716	420	242	159	153	210	317	467	653	869



