SUPRMENU

5.1

This software is provided as-is.
There are no warranties, expressed or implied.
Full disclaimer is found at the end of this doucment.

NOTE
====

This document is subject to change as SUPRMENU develops
and as more detail is added.

Table of Contents

SUPRMENU 5.1 - Menu Program for MS-DOS

SUPRMENU is a front-end menu program which will provide a user-friendly interface to the application and utility programs on your MS-DOS computer system. It is set up so that it can easily be used on a NOVELL network system as well.

SUPRMENU features:

- 0 K memory used when applications are running

- Supports a mouse

- A unique combination of ease of use with powerful control

- Concise, yet easy to learn, syntax for commands for:
  - Popup submenus
  - Dialog boxes
  - File picklist
  - User entry of parameter values
  - Retrieving DOS environment variables
  - Rebooting the computer
  - Logging out of a network

- Dynamic creation of BAT commands to run applications

- Screen saver which runs user-defined program

SUPRMENU is more than a simple menu system which issues one command when a selection is made. SUPRMENU issues a programmable stream of DOS commands (via a dynamically-created BAT file) to run your applications.

Registration

The shareware version of SUPRMENU is not limited in any way; try it, test it, use it. After a short trial period (30 days is suggested), you are expected to register your copy of SUPRMENU by sending a registration fee as indicated on the enclosed registration form.

Upon receipt of registration, a disk with the most recent version of SUPRMENU will be mailed to the you. This diskette will contain other bonus software, as well.

For your convenience, a registration form is included at the end of this document.

Support

Support is provided to registered users. At the time this document was created I could be reached on COMPUSERVE (id: 70142,1324) or on PC-Library (612-435-1785 Userid: Drich).

Users who obtained SUPRMENU from Polich, Frogner & Associates may contact them for support.

Installing SUPRMENU

In order to install SUPRMENU on your system, do the following:

1. Install the SUPRMENU files on your hard disk

   - The SUPRMENU distribution diskette comes with an installation program. All that is necessary to install SUPRMENU is to place the diskette in the A: or B: drive and enter A:INSTALL or B:INSTALL. You will be asked which subdirectory to install SUPRMENU into. After the software is copied to your hard disk, you will be placed in the SMSETUP configuration program (see below).

   - If you wish to manually install SUPRMENU, you should follow these instructions:

     a. Create a subdirectory on your hard disk (suggest C:\SUPRMENU although anything is OK, SUPRMENU adjusts).

     b. Copy the following files to that directory (or extract them from SUPRMENU.LZH, using LHA)

        SUPRMENU.EXE - The menu program.

        SMSETUP.EXE - The setup program which will create the SMMENU.BAT file used to call SUPRMENU. SUPRMENU must ALWAYS be started using this BAT file; therefore, the SMMENU.BAT file should be placed in a subdirectory accessible on the PATH.

        SAMPLE.MNU - Some examples of menu items

        DEMO.BAT
        DEMOMENU.MNU - Demonstration menu system - execute demo by entering DEMO (after running SMSETUP)

DEMO

SAVER.EXE - Program which can be used for the screen saver. SAVER2.EXE is an enhanced version.

SMSAVER.EXE - Program which runs random screen savers. See SMSAVER.DOC.

SMCONV5.EXE - You need to run this to convert a previous level of SUPRMENU to any 5.0 or above level.

REBOOT.EXE - Used in a SUPRMENU-created BAT file where processing must happen first and the internal !REBOOT command cannot be used

c. Change to the subdirectory containing the above files and run SMSETUP (see description below). After specifying subdirectories and colors choose SAVE to write your SMMENU.BAT file.

2. Create SUPRMENU.MNU (or SMMENU.MNU) for your own needs for a main menu (see the instructions below).

3. Create any other menu files you desire.

The programs SUPRMENU.EXE and SMSETUP.EXE (as well as SAVER.EXE, if you use it) must all reside in the same subdirectory. All menu files (SUPRMENU.MNU and any others you define) must reside in this same subdirectory.

On a network, the subdirectory containing SUPRMENU.EXE and SMSETUP.EXE must be available to all users with read/execute access. Anyone changing menu files must have write/delete access as well.

CONVERSION TO SUPRMENU Version 5.0 or above

SUPRMENU Level 5.0 introduced the ability to position the main menu on the screen. This has necessitated a change in the call line for the SUPRMENU.EXE program itself. Since this program is only called from the SMMENU.BAT file, only that file needs to be modified.

For users converting from levels of SUPRMENU prior to 5.0, you must do one of the two following things:

1. Install your update from the distribution diskette using INSTALL. This automatically invokes the conversion program.

2. After manually extracting files from SUPRMENU.LZH on the diskette, run SMCONV5.EXE to convert files.

The conversion program will change the SMMENU.BAT found in the PATH on the machine it is executed from. If you have several workstations on a network, each with its own SMMENU.BAT file, you must execute SMCONV5.EXE on each workstation (it can be executed from a network subdriectory , however). This is a rare situation.

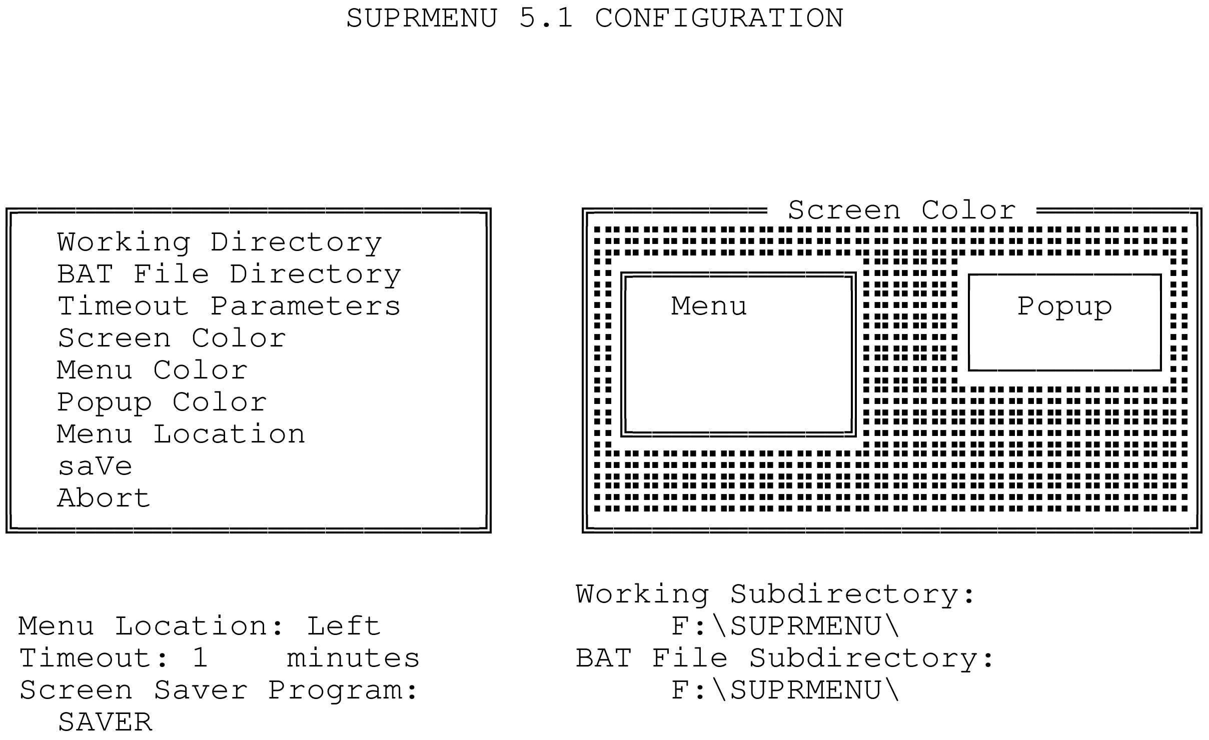
The conversion will default menu positioning to the Left of the screen, as it has been in the past. To change the menu positioning, use SMSETUP.EXE.

RESTRICTIONS (Maximums & Minimums)

Minimum DOS level required:	3.x
Characters in a menu title:	30
Characters in a command line:	80
Characters in a submenu title:	30
Number of menus in a menu defintion file:	Unlimited
Number of total command lines in a menu:	600
Items on a menu:	15
Items in a submenu:	10
Items in a dialog box:	5 max (75 char max, see below)

Configuration - Running SMSETUP

SMSETUP is the program supplied to allow you to interactively create the SMMENU.BAT file required to run SUPRMENU. When you execute SMSETUP, you are presented with this screen:



```
                    SUPRMENU 5.1 CONFIGURATION


  Working Directory               Screen Color
  BAT File Directory
  Timeout Parameters            Menu            Popup
  Screen Color
  Menu Color
  Popup Color
  Menu Location
  saVe
  Abort

                               Working Subdirectory:
  Menu Location: Left               F:\SUPRMENU\
  Timeout: 1   minutes         BAT File Subdirectory:
  Screen Saver Program:             F:\SUPRMENU\
    SAVER
```

The menu options are as follows:

Working Directory - This is the subdirectory where SUPRMENU will write the temporary BAT file which is created from the commands you place in the menu file. This BAT file is called from SMMENU.BAT to execute your program.

You can place all user BAT files in the same subdirectory as SUPRMENU or any other subdirectory to which all users are granted access. SUPRMENU creates a unique BAT file each time the user picks a menu item. The name of the BAT file is based on the value of the DOS SMSITE variable if it exists. Its name is found in an environment variable called MENURUN and used from SMMENU.BAT to run the unique BAT file.

In any case, all BAT files for running menu selections will be placed in the subdirectory named on this selection.

On a network, all users must have write/read/delete access to this subdirectory. This may be the same subdirectory as SUPRMENU.EXE, unless you want to restrict access to the EXE files, in which case you might create a special subdirectory just to contain these BAT files.

BAT File Directory - This is the subdirectory where you want SMSETUP to place the SMMENU.BAT file which will start SUPRMENU. This subdirectory should be on your PATH statement or be the directory which contains SUPRMENU.EXE. If it resides with SUPRMENU.EXE, it could be called using an explicit path (see below under Executing SUPRMENU), since SUPRMENU adjusts to whatever subdirectory it is in.

On a network, since the SMMENU.BAT file is never modified dynamically, only the person who creates it with SMSETUP needs full access. Ordinary users need read access to this subdirectory. It is suggested that SMMENU.BAT be placed in a subdirectory on every user's path, such as PUBLIC.

Timeout Parameters - Enter minutes of inactivity until screen saver program is invoked; 0 means do not use screen saver. You also enter the command to execute.

Screen Color - This is the background color for the menu screen.

Menu Color - This is the color of the main menu.

Popup Color - This is the color of popups for SUBMENU, CHOICE, EXIT, etc. commands.

Menu Location - Select whether the main menu window will appear on the Left, Center or Right side of the screen. This also affects the position of popup items. Note that this will NOT change the small display in SMSETUP.

Save - SMSETUP will write a new SMMENU.BAT file to reflect your changes and place it in the "BAT file directory".

Abort - Exit SMSETUP without writing a new BAT file.

The SMMENU.BAT file written by SMSETUP will contain a line starting with :CONFIG. This line, which is ignored by DOS, contains the configuration you defined.   Note that SMSETUP will attempt to find the existing SMMENU.BAT file and read the items on the :CONFIG line in that file.

If you define a screen saver timeout, the SMMENU.BAT file will contain a line starting with :SAVER which defines the timeout parameters to SMSETUP.

SMTEST.BAT

SMSETUP also creates a file in the SUPRMENU subdirectory called SMTEST.BAT. This file works the same as SMMENU.BAT, but will not execute the commands created; instead, it lists them to the screen and returns to SUPRMEN after you press a key. This BAT file is intended for use in testing SUPRMENU menu files.

To use SMTEST.BAT:

1. Change to the SUPRMENU subdirectory

2. Be sure there is a SUPRMENU.MNU or SMMENU.MNU file created.

3. Execute SMTEST

Executing SUPRMENU

After you have created a SUPRMENU.MNU (or SMMENU.MNU) file and have run SMSETUP, you execute SUPRMENU by simply entering:

```
            SMMENU
```

at the DOS prompt. (Note that SMMENU is a BAT file).

If you wish to execute a MNU file with another name, for example BILL.MNU, enter the command

```
            SMMENU BILL    ( assumes BILL.MNU is in the SUPRMENU
                              subdirectory)
```

where the parameter is the filename (MNU is assumed for an extension if none is given). This would allow separate menus for different users on a network.

Monochrome Monitors

Normally, SUPRMENU correctly detects the kind of video card you have and adjusts. If you are driving a monochrome monitor with a CGA card or some other unusual circumstance, you can force SUPRMENU to run in monochrome mode by setting an environment variable:

```
SET SMCOLOR=MONO
```

This could be done in AUTOEXEC.BAT or elsewhere.

Rodents

A mouse can be used to point-and-shoot the desired menu selection, or you can use the cursor keys and RETURN.

Creating menu files - Menu File Format

The SUPRMENU release includes sample menu files. You are encouraged to print them and examine them, they have extensive comments which describe each menu item.

Defining Menus

A menu file (default is SUPRMENU.MNU or SMMENU.MNU) contains the definitions of one or more menus which each appear in the window on the left side of the screen when selected. Each menu has a name, or "label"; this name is how the menu is found within the menu file. The menu name is preceded by an asterisk.

```
    *menuname
```

The default menu is the menu labeled *MAIN. SUPRMENU will find this menu and display it when first executing (unless another menu is specified in the second parameter on the call to SMMENU.BAT).

The menu label can also define the menu to which the user will return if they press ESC in the defined menu. In order to specify this, add a comma and the "return" menu name/label:

```
    *menuname,return_menu_name
```

If a comma and return_menu_name are not given, the default is to return to *MAIN when the user presses ESC.

You may also specify which menu item is highlighted on the menu returned to by appending a comma and item number as the third field on the menu definition:

```
    *menunuame,return_menu_name,item_number
```

The first line after the menu label is the text which will appear as the menu title in the top border of the menu window.

Therefore, a set of menus are defined as:

```
            *MAIN
            title
                 .
                 .   (Choices go here, see below)
                 .
                 .
             *menuname,rtnmenu
             title
                 .
                 .
                 .
             *menuname,rtnmemnu
             title
                 .
                 .
                 .
                etc.
```

Menu Items

Each menu will contain up to 15 choices, or menu items. These choices are defined after the "*menuname" and "title" lines.

Each menu item, or choice, is defined by

1) A line which begins with a '#' character and defines the hot key which can be used to select the item and the text to be displayed in the menu window.

```
#h,description
^     ^
|     |_____________ Will appear on the menu, up to 30 chars
|___________________ The hotkey to execute selection, this
                      character must appear in the
                      description
```

The hot key can be empty, for example:

```
#,Item Description
      -or-
# ,Item Description
```

If the hot key is empty, no hot key will be defined for the menu item.

2) Command lines following this which define:

a) DOS commands
These are written to a BAT file, in sequence after any %1...%9 variables are replaced with values from SUPRMENU commands.

b) SUPRMENU commands
These are special commands, which all begin with an exclamation (!) mark. These commands are evaluated in sequence with the other commands; they are defined below.

c) Password
A password preceded by a caret (^) can defined. The user must enter the password before the menu selection is executed.

Notice that leading spaces are ignored for all lines in the menu file and blank lines are skipped when reading menus.

SUPRMENU commands

When a menu item is selected, SUPRMENU reads each of the command lines for that menu item IN SEQUENCE. If a line is one of the following commands, the action specified is taken, i.e. get popup selection, etc. If the line is not one of the following commands, it is assumed to be a DOS command.

Many of the SUPRMENU commands listed below place a text string in one of 9 "memory variables" which are then designated in commands as %1 ... %9. These variables are the means by which user input is inserted in the commands which execute the menu selection.

The following SUPRMENU commands allow the use of the %n variables to dynamically change parameters (see the command descriptions for details): !MSG, !ASK, !CHOICE, !FILE, !EXEC.

DOS commands first have all %1 to %9 codes replaced with whatever is placed in those memory variables by the commands below (note that these are replaced in numeric sequence, first %1, then %2, etc.). The DOS command is then written to the "MENURUN" BAT file (see below under "How SUPRMENU works" if you want more details about the "MENURUN" BAT file creation).

SUPRMENU Command Descriptions:

Comments:

;text

This is a comment and is skipped. Comments appear on lines by themselves and can appear anywhere.

Password:

^password

Password protect the entry. If the entry is a !MENU, !REBOOT, !LOGOUT or !EXIT entry, this must be the first item after # line. This password IS NOT case-sensitive.

Go to another menu in the current menu file:

!MENU,menuname

Read another menu in the current menu file and display it. Any other commands for the current menu item are ignored, except for password as described above. Note that "menuname" should NOT include the leading asterisk.

When SUPRMENU is first executed it looks for a file called SUPRMENU.MNU or SMMENU.MNU (or the file given on the call line) and looks in that file for the menu labeled *MAIN (or the name given on the call line).

Load another menu file:

!LOAD,filename

Load another menu file and display the menu labeled *MAIN. If the filename has no extension, .MNU will be assumed.

Exit SUPRMENU:

!EXIT

Issuing this command will cause the Exit to DOS dialog box to appear. Any other commands for the menu item are ignored, except for a password as described above.

Display a message:

!MSG,text

Displays a one-line message box with the "text". Text can be up to 32 characters long. The "text" parameter can contain %n variables, which will be substituted before display.

Erase an existing message:

!NOMSG

Closes !MSG box. This is only needed if there is further interaction with the user for which the message shouldn't be displayed. The message box is automatically closed at the end of processing of menu commands, just before the "MENURUN" BAT file is invoked.

Issue LOGOUT command and exit:

!LOGOUT

This command is useful for NOVELL networks, although it could be adapted to other uses via a LOGOUT.BAT file. When this command is executed, SUPRMENU clears the DOS keyboard buffer and stuffs "logout" plus a CR into the buffer. It then exits to DOS, which will pick up the command and execute it. Any other commands for this item will be ignored, except for password protection as described earlier. Note that this is a DOS-sensitive command; it works at all levels of DOS tested, but you should test it with the level of DOS you are using (at worst, it just won't work).

Reboot the computer:

!REBOOT,W
!REBOOT,C

These commands will cause the computer to reboot. The W option forces a warm boot (similar to Ctl-Alt-Del), the C option forces a cold boot (as when turning the computer on).

Execute DOS command directly

!EXEC,cmd — This command causes the DOS command "cmd" to be executed immediately from within SUPRMENU during the creation of the MENURUN BAT file. The command DOES NOT become a part of the BAT file, but is executed via a temporary shelling to DOS at the point that the !EXEC command is encountered while building the MENURUN BAT file. The cmd parameter can contain %n variables.

Notes:

1. SUPRMENU is not terminated during the execution of the command on the !EXEC command. Beware of the following:

   a. About 150K - 160K of memory will be occupied by SUPRMENU and multiple invocations of COMMAND.COM.
   b. SUPRMENU has inserted itself in the interrput handling for keyboard input in order to properly handle Ctrl-C and Ctrl-ALt-Del. This could affect some programs.
   c. Loading or unloading TSRs could cause problems including locking up the machine and, less likely, destroying data.

2. The most common use of this command is to issue DOS internal commands, like COPY, or small utility programs to perform actions. It is most useful as part of a menu item that will do !REBOOT, !LOGOUT or some such command. Note that most things can be done in the normal way via the BAT file produced by SUPRMENU.
3. DO NOT use this command as a substitute for the normal process of creating a BAT file and executing your programs from there.

The following commands all have a number from 1 to 9 as the second item ("n"). There are 9 internal string variables defined which are used for "parameter substitution" in the commands generated. The construct %1 ... %9 will have the current value of the corresponding variable substituted as the BAT file is created.

Ask for user parameter input:

`!ASK,n,prompt`

A box with "prompt" will be displayed and the user will be able to enter a string of up to 48 characters. The user's response will be placed in the designated (1 .. 9) internal string. This string will be used in parameter substitution. ESC will abort the current menu selection.

The "prompt" parameter can contain %n variable references which are substituted before displaying the prompt.

Prompt may be up to 45 characters long.

Dialog box with choice buttons:

`!CHOICE,n,choice,choice,.....,prompt`

A dialog box will be displayed with "prompt" displayed and the choice strings displayed as horizontal dialog box selections. Whichever choice string the user selects will be placed in the designated internal variable and will be used in parameter substitution. ESC will terminate the current menu selection.

The "prompt" parameter can contain %n variable references which are substituted before displaying the prompt.

There may be up to 5 choices. The total length of the choices including 2 spaces between them cannot exceed 75; Likewise the prompt string cannot exceed 75 characters in length.

Popup submenu:

`!SUBMENU,n,submenuname`

A popup menu, as defined at *submenuname, will be displayed. This a vertical, scrolling menu with point-and-shoot selection. The variable will receive the return string for the selected item as defined in the submenu definition (see below for format.) If ESC was pressed, the current menu selection will be aborted. There may be up to 10 items in a submenu.

Retrieve environment variable:

!GETENV,n,varname Put value of the Environment variable "varname" into variable # n. If the Environment variable isn't found, variable # n will be 0 length.

Select a file from a list:

!FILE,n,fmask[,default] Put up a picklist of filenames from a subdirectory which match the filemask passed (which may include wildcards). The user can pick one and press enter, in which case variable #n will contain the full path and filename. If the user presses ESC, variable #n will receive the default if specified, or the current menu selection will be aborted if no default is specified.

The "fmask" parameter can contain %n variable references which are substituted before gathering the list of files.

Clear a SUPRMENU variable:

!CLEAR,n Clears the designated variable to a 0 length string.

Set a SUPRMENU variable:

!SET,n,value Sets the variable #n to the string "value".

The following two SUPRMENU commands control the execution of commands and creating of the "MENURUN" BAT file. The optional "nl" parameter on each of these commands defines the number of lines to skip if the command evaluates as TRUE. The possible values of "nl" are:

```
      not given   - Skip the next line.

      an integer  - Skip the number of lines specified.  This can be a
                    negative number; but be careful of infinite loops.
                    If the integer attempts to skip beyond the end of
                    the commands for the item, it is the same as END.
                    If a negative integer tries to skip past the first
                    command, it is the same as QUIT.

      END         - Skip the rest of the commands for this menu item
                    and issue the BAT file to run the application.

      QUIT        - Abort processing and return to the menu.
```

Skip line(s) if equal:

```
      !SKIPIF,n,value[,nl]  This command causes lines (either another
                            SUPRMENU command or a DOS command) to be
                            skipped if the parameter variable #n
                            starts with the string "value".
```

Skip line(s) if not equal:

```
      !SKIPNOT,n,value[,nl] This command causes lines to be skipped if
                            the parameter variable #n DOES NOT start
                            with the string "value".
```

Note that the value parameter in the two commands can be null (0 length). The commands then evaluate to TRUE if the value held in the numbered variable is also null. Examples:

```
!SKIPIF,1,
!SKIPNOT,2,,END
!SKIPIF,2,,QUIT
!SKIPNOT,,3
```

Sample menu items:
==================

1. Exiting to DOS - (This can be password protected as shown) (Also can exit to DOS using Shift-F10, this is NOT password protected.)

```
    #Q,Exit to Dos             }  To exit to DOS
        ^password              }    (Password)
        !EXIT                  }
```

2. Call a program and pass user entered parameter

```
    #h,Program Description                 }
          cmd                              } Call a program,
          cmd                              }   up to 20 cmd lines
          !ASK,1,Enter a Parameter         }
          cmd %1                           }
```

3. Call several programs in sequence with user specified parameters.

```
   #h,description                             }
    cmd                                       } Call a program,
    !CHOICE,2,choice1,choice2,prompt          } Dialog Box
    cmd %2                                    } Use dialog box response
    !ASK,3,prompt                             } User text
    cmd  %3  %2                               } Substitute the above
    !CHOICE,3,choice2,choice2,choice3,prompt } Dialog Box (notice reuse
    cmd %3 /X                                 }   of variable number)
```

4. Comments in menu file (will not be used)

```
; These are comment lines which will be skipped on reading the
; menu file
```

5. Switch to another menu defined in same menu file.

```
  #h,description               }  This format calls
      !MENU,menuname           }      another menu in
                               }      current menu file
```

6. Switch to another menu file

```
  #h,description
      !LOAD,filename           } Get a new menu file and load
                               }   the menu called *MAIN
```

7. Return to Main Menu (could be first item - see SAMPLE.MNU)

```
  #!,! Return to Main menu
      !MENU,MAIN
```

Note that this is also accomplished when the user presses ESC while in a menu.

8. Reboot computer routine

```
  #R,Reboot Computer
      ^secret
      !CHOICE,1,WARM,COLD,Type of Boot
      !SKIPIF,1,WARM
      !REBOOT,C
      !REBOOT,W
```

9. Using the !ASK command

```
  #P,PC Shell
     !ASK,1,Enter a subdirectory name
     PCSHELL %1
```

10. Using the !CHOICE command, putting up a dialog box

```
  #F,Format a diskette
      !CHOICE,2,A:,B:,Diskette Drive
      format %2
```

11. Using the !SUBMENU command, displaying a popup menu of choices

    (See above under SUBMENU format).

12. Using the !GETENV command to retrieve a DOS Environment Variable

```
  #R,Run this program

      !GETENV,2,USERNAME
      CD \DATADIRS\%2
      PROG
```

13. Using the !FILE command

```
   #E,Edit a file

      !MSG,Select file to edit
      !FILE,3,D:\DOCS\*.DOC
      Edit %3
```

14. Defining an item which does not have a hotkey

```
   #,List a File

      !MSG,Select file to list
      !FILE,4,D:\DOCS\*.DOC
      LIST %4
```

15. Handling re-configuration and reboot.

```
   #C,Configuration

      !CHOICE,1,WINDOWS,DOS,DESQVIEW,Choose a configuration
      !EXEC,Copy C:\CONFIGS\%1.BAT C:\AUTOEXEC.BAT > NUL
      !EXEC,Copy C;\CONFIGS\%1.SYS C:\CONFIG.SYS > NUL
      !REBOOT,W
```

SUBMENU format

SUBMENUs are popup menus created by the !SUBMENU command. They are defined in the Menu File which contains the !SUBMENU command which invokes them.

Submenus are located and referred to by a name or label, much as menus are. A submenu definition:

1) Begins with a line containing the submenu name preceded by an asterisk. This is the line searched for and found when SUPRMENU encounters a !SUBMENU command.

2) The next line defines the title of the submenu window.

3) Up to 10 submenu items of the form:

   ```
   display string=return value
   ```

   The display string will appear as a choice in the submenu popup. When a submenu item is selected, SUPRMENU places the "return value" into the SUPRMENU variable (1-9) designated on the !SUBMENU command. This can be used in subsequent commands in the menu item.

Submenu format summary:

```
          Line 1:  Submenu title
     Line 2 to n:  display value=return value
        <line with * , or EOF terminates>
```

Note that, if ESC is pressed the variable designated on the !SUBMENU command will be made zero length.

Sample of submenu

The submenu is called by:

```
!SUBMENU,2,PRSELECT
```

The submenu is defined elsewhere in the MNU file as:

```
*PRSELECT              searched for on !SUBMENU item
Select Printer
    IBM GRAPHICS=IBMG   } string to display=string to return if
    EPSON=EPSON         }                  selected
    LASER=LASER

  etc. up to 10 submenu items
```

Executing SUPRMENU

To execute SUPRMENU,

1. If SMMENU.BAT was placed in a subdirectory on the PATH, enter

```
SMMENU
```

2. If SMMENU.BAT was placed in a subdirectory NOT on the PATH, enter

   a. CD method

```
CD  d:\path_to_subdir
SMMENU
```

   b: Explicit path method

```
d:\path_to_subdir\SMMENU
```

Normally, when first executing, SUPRMENU looks for the default menu file (SUPRMENU.MNU or SMMENU.MNU) in the SUPRMENU.EXE subdirectory and then shows the *MAIN menu. You may override these defaults by placing parameters on the call to SMMENU:

```
SMMENU  [menufile  [start_menu ]]

 where:
        menufile is the name of the menu file to use.
                 If no extension is given, .MNU is assumed.
        start_menu is the name of a menu in the menufile
                 to start at.  If not given, MAIN is assumed.
```

Automatically generated commands

Note: It is not necessary to enter a return to SUPRMENU as part of the commands for a menu item, as SUPRMENU will append these commands to the BAT file created. In addition, SUPRMENU puts @ECHO OFF at the beginning of the file, so you don't need that command either. This file is named is named SM.BAT (or its name is SMcccccc.BAT where cccccc is the value of the SMSITE DOC environment variable) and is created in the working subdirectory named on the call to SUPRMENU. The name of the BAT file created is put in the MENURUN DOS environment variable which is used by SMMENU.BAT to execute the commands created (this is done by a command in the SMMENU.BAT file).

The format of this BAT file is:

```
@ECHO OFF
cmd         }
cmd         }  Commands from your MNU file
cmd         }
.           }
.           }
.           }
C:                       } Added by SUPRMENU:
CD \SUPRMENU              } drive, subdirectory and menufile will

SMMENU MENUFILE MENUNAME  } vary according to where
                         } SUPRMENU is  installed and the
                         } current menufile and menu
```

BAT files and the CALL command

One problem you may experience is that your set of commands to run your application may terminate without returning to SUPRMENU. This is usually because one of the commands you have issued runs a BAT file.

Whenever you invoke a BAT file from another BAT file, DOS will not return you to the "parent" BAT file when the second ("child") BAT file completes. DOS 3.? and above contain a CALL command which calls the second BAT file as a subroutine, which means DOS returns to the calling BAT file when the called BAT file completes. This is, therefore, not a deficiency of SUPRMENU but simply a normal DOS behavior.

If your application call is to a BAT file, instead of putting

```
      DOAPP            (which invokes DOAPP.BAT)
```

as a menu action command, use

```
      CALL DOAPP
```

Remember that SUPRMENU is creating a BAT file which is executed to run your application. You need the CALL so that control returns to that BAT file, which will then restart SUPRMENU.

Note: For levels of DOS which do not support the CALL command, use the following syntax:

```
           COMMAND/C DOAPP
```

SMSTART/SMRUN/SMEXIT

As of level 5.1, SUPRMENU has the capability to run these external programs when starting, when running it 'MENURUN' BAT file and when EXITing to DOS.

When SUPRMENU is executed, the first time or any time it is returned to after execution of an application, it checks for the existence of SMSTART.COM, SMSTART.EXE or SMSTART.BAT in the SUPRMENU subdirectory. If SMSTART is found, it is executed after the menu is displayed, but before the highlight appears for user selection. This could be used to put something on the screen, check special security, etc.

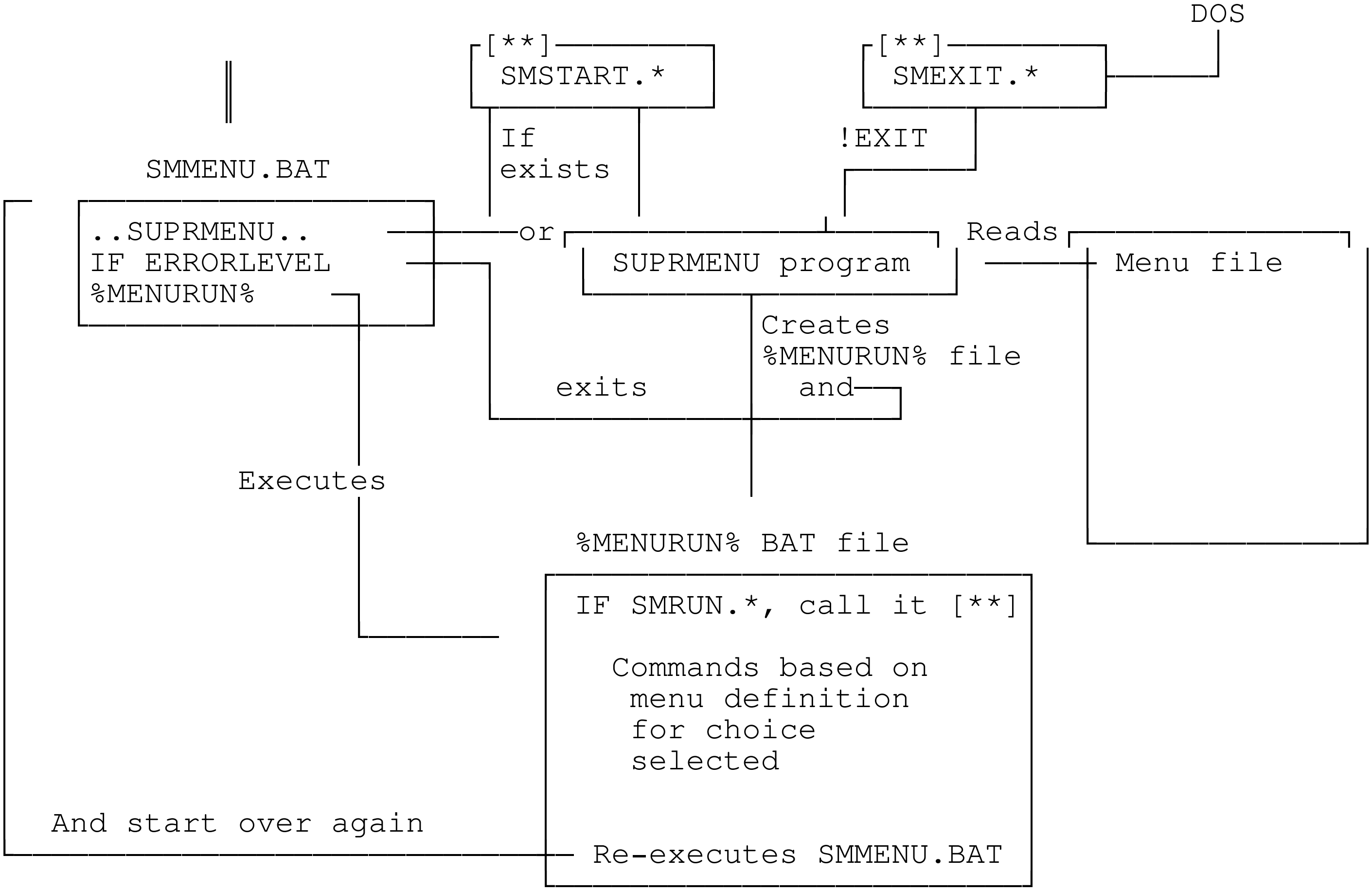
Whenever a menu item is selected and a 'MENURUN' file is created and executed, the first command in the 'MENURUN' file is

```
    IF EXIST SMRUN.* SMRUN
```

which checks for existence of SMRUN.COM, SMRUN.EXE or SMRUN.BAT in the SUPRMENU directory and runs it if it exists. This will happen whenever a menu item is selected which creates the 'MENURUN' BAT file. This could be used to change to a user directory, log usage, etc. Use this for something which executes for ALL menu items; programs for individual items should be placed in 'MENURUN' by commands in the menu definition.

Whenever the user exits SUPRMENU from the "Exit to DOS" dialog invoked by !EXIT, SUPRMENU checks for the existence of SMEXIT.COM, SMEXIT.EXE or SMEXIT.BAT in the SUPRMENU subdirectory and, if found, executes it just before terminating. Note that, at the time this is executed, the screen has NOT been cleared and SUPRMENU is still registered to handle the keyboard interrupts, i.e. the user will still not be able to Ctrl-C or Ctrl-Alt-Del.

Technical Description



[**] These files executed only if they exist.

How SUPRMENU works

In order for SUPRMENU to use 0 K of RAM when an application is started from SUPRMENU, it uses BAT files; one BAT file to run SUPRMENU and another BAT file, created dynamically by SUPRMENU from the menu action lines, which runs the application and then restarts SUPRMENU.

The file SMMENU.BAT is executed to start SUPRMENU. SMMENU.BAT is created in the "BAT file subdirectory" by SMSETUP. The first part of SMMENU:

- Changes to the SUPRMENU subdirectory:

```
        F:                          (example lines)
        CD  \MENU
```

- Then executes SUPRMENU.EXE

```
        SUPRMENU 31 48 61 L F:\MENU %1 %2
```

When SUPRMENU first executes, it checks for SMSTART.BAT/.COM/.EXE in the SUPRMENU subdirectory. If it exists, it is executed just after the menu is drawn.

Now SUPRMENU is executing and you can select a menu item.

When you select a menu item, SUPRMENU evaluates the menu action lines, creating a file called SM.BAT (or, see alternate method using the SMSITE DOS Environment Variable) in the "working subdirectory" (defined in SMSETUP).

If the menu action is !EXIT, !MENU, !LOGOUT, !REBOOT, or !LOAD a BAT file is not created.

Once the SM.BAT file is written, SUPRMENU terminates with an ERRORLEVEL of 0, unless the !EXIT command was found, in which case it terminates with an ERRORLEVEL of 99.

The SMMENU.BAT file regains control and:

- Checks for !EXIT (or an error)

```
IF ERRORLEVEL 99 GOTO END
```

- Checks for the alternate BAT file method and adjusts automatically

```
SET MENURUN=SM%SMSITE%.BAT
```

- Chains (without return) to the BAT file that SUPRMENU created.

```
%MENURUN%
```

- This label only reached from IF ERRORLEVEL above

```
:END
```

The dynamically-created BAT file created by SUPRMENU ends with:

```
F:                           ( SUPRMENU drive)
CD \MENU                     ( SUPRMENU subdirectory)
SMMENU last_menu_file  last_menu_label
```

thereby restarting SUPRMENU.

This all may sound complex and quite daunting. All you really need to know is:

1. Put SUPRMENU.EXE and SMSETUP.EXE in the same subdirectory.
2. Create SMMENU.BAT using SMSETUP
3. Create a menu definition file (SUPRMENU.MNU or SMMENU.MNU are the defaults) which also must be in the same subdirectory as SUPRMENU.EXE.
4. Run SUPRMENU using SMMENU.BAT

If you use the alternate interface method in a network environment:

5. Be sure that the DOS environment variable SMSITE is set from your AUTOEXEC.BAT

   ```
   SET SMSITE=id
   ```

   or from a NOVELL login script

   ```
   DOS SET SMSITE="%STATION"
   ```

CALL to SUPRMENU (MUST be issued from SMMENU.BAT)
===

```
SUPRMENU scrncolor mnucolor popupcolor wrkng_dir mnuloc [mnufile [mnuname]]
```

Sample SMMENU.BAT (for Networks)
This MUST be created by SMSETUP.
This is only for your interest.
================================

```
@echo off                               │ -No echo of commands
REM Do not delete or change the next two lines
:CONFIG 63 48 112 D:\MENU\              │ -SUPRMENU configuration
:SAVER 1;SAVER                          │ -Screen saver configuration
SET SMSAVER=1;SAVER                     │ -Screen saver values for SUPRMENU
F:                                      │ -Change to drive and directory
cd F:\SUPRMENU                          │     containing SUPRMENU and menus
SUPRMENU 63 48 112 D:\MENU L %1 %2      │ -Execute SUPRMENU
IF ERRORLEVEL 99 GOTO END               │ -Returned by ABORT or Exit to DOS
SET MENURUN=SM%SMSITE%.BAT
D:\MENU\%MENURUN%                       │ -Run the BAT file created
:END                                    │ -Come here on abort or exit
```

Alternate methods to interface to applications:

Normal method:

Normally, when you select a menu item, SUPRMENU creates a BAT file with the name SM.BAT in the "working subdirectory." It then places the commands in that file.

Alternate method:

If your SUPRMENU system is to be used by multiple users on a network, you will have problems with SUPRMENU overwriting the SM.BAT file when a second user runs an application while the first user is still executing an application. SUPRMENU provides an alternate method of determining the BAT file name which can provide a unique BAT file name for each user. In order to use this method, you must define an environment variable SMSITE which is equated to a 1 to 6 character identifier which will be used in naming the BAT file created by SUPRMENU. In a standalone environment, this will normally be set in your AUTOEXEC.BAT file; in a Novell network environment this could be done in AUTOEXEC.BAT or in the login script. On a network, each workstation must have a different identifier defined.

In AUTOEXEC.BAT:

```
SET SMSITE=DDDD
```

In login script:

```
DOS SET SMSITE="%STATION"
```

When SUPRMENU detects the existence of this environment variable, it names the BAT file "SM" plus the SMSITE string; for example, if SMSITE is set to "XYZ", the BAT file will be called SMXYZ.BAT. The SMMENU.BAT file created by SMSETUP has commands which will automatically accommodate this alternate method.

What you need to know:

All of this may sound confusing, so here are the basic things you need to know:

Normal method (try this first);

1. Install SUPRMENU.EXE, SMSETUP.EXE in a subdirectory.
2. Run SMSETUP and define where SMMENU.BAT is to be written and where to write the SM.BAT files.
3. Create a SUPRMENU.MNU file in the same subdirectory as SUPRMENU.EXE.
4. That is all you need. Just enter SMMENU to run SUPRMENU.

Alternate method (if you are in a multi-user, networked environment):

1. Everything is set up as above.
2. Add the SET SMSITE= command to your AUTOEXEC.BAT or login script.
3. That's all!!

Networks

What you need to check if SUPRMENU is run on a Novell Network:

1. SMMENU.BAT is in a subdirectory (PUBLIC is suggested) to which all users have at least read/execute access. The person running SMSETUP must also have write/delete access.

2. SUPRMENU.EXE, SMSETUP.EXE, SAVER.EXE and REBOOT.EXE, as well as any menu files (SUPRMENU.MNU/SMMENU.MNU and others you define) must all reside in the same subdirectory. All users must have read/execute access to this subdirectory. Anyone changing the MNU files requires write/delete access.

3. A subdirectory is designated to SMSETUP as the "working" subdirectory. It is here that SUPRMENU writes the BAT files used to run your applications. Using the SMSITE variable, each user will have a unique BAT file name allocated for that user. All users must have full rights to this subdirectory, as files are created and destroyed.

4. In the system login script, include

   ```
   DOS SET SMSITE="%STATION"
   ```

   This will give each user a unique BAT file name in the working subdirectory. Note that %STATION must be in all UPPER CASE or it won't work.

5. Also in the system login script, you should place

   ```
   EXIT "SMMENU"
   ```

   at the end to automatically start up SUPRMENU for everyone. This can also be in any user login script.

Tricks

Nested replacement variables

Since SUPRMENU evaluates and replaces variables in numeric order (that is, %1, then %2, then %3, etc.), you can play a trick like the following:

In a menu item definition:

```
          !SUBMENU,1,PRSELECT      | Returns ENDCAP or NETWORK in %1
          !SKIPIF,1,ENDCAP         | Skip next if %1=ENDCAP, i.e. local
          !SUBMENU,1,NETPRSEL      | Find out which network printer
                                   | Note that returned has %2 in it
          !SKIPIF,1,ENDCAP         | Don't do next for local printer
          !SUBMENU,2,FFSEL         | Get form feed choice into %2!!!
          %1                       | First replaces %1
                                   |  If it is network, resulting string
                                   |     contains a %2
                                   | Next the %2 is replace with FF or NFF
                                   | Neat, huh?
```

Submenus for above:

```
          *PRSELECT
          Select A Printer
          Network Printer=NETWORK
          Local Printer=ENDCAP

          *NETPRSEL
          Network Printer
          Laser Printer=CAPTURE NB NTT %2 Q=PRINTQ_1
          Dot Matrix Printer=CAPTURE NB NTT %2 Q=PRINTQ_0

          *FFSEL
          Send Form Feed
          Form Feed=FF
          No Form Feed=NFF
```

Menu calls while creating BAT file

When converting from other menu systems, it may be necessary to call another menu in the middle of creating a command stream to call your application. The idea would be to insert several commands in the middle of the command stream depending on a user choice.

First we have the menu item, CHOICE_X. Within the commands generated when this item is selected we have a variable set of commands, based on user selection on the submenu CHOICE_X_S.

```
      #X,CHOICE_X

           ..command..
           ..command..
           ..command..
           !SUBMENU,2,CHOICE_X_S
           !SKIPNOT,2,CH1,4
           ... command if choice 1 ..
           ... command if choice 1 ..
           ... command if choice 1 ..
           ... command if choice 1 ..
           !SKIPNOT,2,CH2,5
           ... command if choice 2 ..
           ... command if choice 2 ..
           ... command if choice 2 ..
           ... command if choice 2 ..
           ... command if choice 2 ..
           !SKIPNOT,3,CH3,2
           ... command if choice 3 ..
           ..command..
           ..command..
           ..command..
           ..command..

   (later in the menu file, submenu CHOICE_X_S)

    *CHOICE_X_S
    A submenu title
    This is choice 1=CH1
    This is choice 2=CH2
    This is choice 3=CH3
```

Other tricks:
============

If you find other neat tricks, let us know. We will include them in future documentation with an attribution and thanks to you.

Appendix A. Error Messages

The following error messages may be displayed by SUPRMENU under the circumstances indicated. SUPRMENU tries to catch the most obvious errors, but some things you do in the menu file may not generate error messages even though they don't work.

1: Error reading menu file - Initial I/O

This error is generated on the first attempt to read a line from the SUPRMENU menu definition file. It generally means that the file is empty.

2: EOF encountered in menu file searching for menu

The menu which was referenced by a !MENU command or on the call to SUPRMENU (this might be MAIN if no menu file name is given on the call to SUPRMENU) was not found. SUPRMENU searches the current menu definition file from beginning to end for a menu. The menu name must be on a line and be prefixed with an asterisk. The most common causes of this error are that you didn't define a menu which is called or your menu file does not contain *MAIN.

3: Error opening menu file - cannot find SUPREMNU.MNU or SMMENU.MNU

SUPRMENU has tried to open each of these files in turn and has not been successful. This normally means you have not defined a menu file or you are not in the correct subdirectory. If you use the SMMENU.BAT file generated by SMSETUP and have created one of the above files, you will not get this error.

4: Too many items in menu - only 15 allowed.

You have defined too many menu items (defined by # character, see above). Only 15 are allowed on any menu. You should split up the menu and create menus which are called by options on the current menu (i.e., create a menu tree).

5: Menu item definition (#) does not have comma!

A menu item is defined as #c,text where "c" is the hotkey (it can be blank) and text is the text displayed on the menu. These are separated by a comma. Usually this error is caused by a typo.

6: More than Maximum number of menu commands - consult docs

There may be a maximum of 450 (small model) or 600 (large model) commands in all of the menu items in a menu. These include both SUPRMENU commands and the commands to be written to the SM.BAT file. They may be distributed amongst menu items as you desire but total number of commands cannot exceed the numbers given above.

7: Command too long

A command line in the menu file cannot exceed 80 characters (leading spaces are not counted as part of the 80 characters).

8: SUBMENU does not have title

The *name which starts a submenu definition must be followed by a line containing the title for the submenu.

9: Parse error (Command parameters= <<command parameters>>)

This is usually caused by not having a comma and a value on a SUPRMENU command. Check the command parameters listed and compare these with your menu file.

10: Submenu - Error opening menu file, initial I/O

This is similar to Error 1, but occurs when searching for a submenu referred to on a !SUBMENU command.

11: Display item too long in SUBMENU definition

The text to the left of the = sign in a submenu item definition will not fit on the screen, i.e. it would extend past the right side of the screen. Make the text shorter.

12: Only 10 items allowed in SUBMENU definition

Only 10 items may be defined in a SUBMENU.

13: Cannot open MENURUN BAT file for writing.

SUPRMENU cannot open the file which it creates to execute your application. This file is named SM%SMSITE%.BAT. This would be caused by a security violation on a network, or the file exists and is read-only.

14: Error on opening Menu Window

This is a SUPRMENU internal error. Call or write the author with a description of what happened.

15: SKIP increment not END, QUIT or a number

A !SKIPIF or !SKIPNOT command contains the optional "number of lines to skip" parameter but it is not an allowed value.

16: Variable substitution makes file mask too long.

In the process of substituting %n variables in a filemask on a !FILE command, the maximum length of a path+file was exceeded.

17: Variable substitution creates string longer than 120 characters.

During substitution of %n variables in a !MSG, !ASK, !CHOICE or !EXEC command, the insertion of the substituted text cause the parameter to be longer than 120 characters, the maximum length that SUPRMENU can handle.

SUPRMENU REGISTRATION

David K. Rich
504 Selby Ave #4
St. Paul, MN 55102

Sold To: ____

Address: ____

____

____ __________

==

____ copies of SUPRMENU at $35 $____________

Total remitted: $____________

Format: ___ 5-1/4" 360K (2 disks) ___ 5-1/4" 1.2 Meg

___ 3-1/2" 720K

Send this form with your check to:

David K. Rich
504 Selby Ave. #4
St. Paul, MN 55102

Thank you for your support

Registered owners will be sent a disk containing the most recent version of SUPRMENU .

If you obtained SUPRMENU from Polich, Frogner & Associates, you are already a registered user.

DISCLAIMER

DAVID K. RICH DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ALL SUCH WARRANTIES ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. NEITHER DAVID K. RICH NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE THE CREATION, PRODUCTION OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE EVEN IF DAVID K. RICH HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR CLAIMS. IN NO EVENT SHALL DAVID K. RICH'S LIABILITY FOR ANY DAMAGES EVER EXCEED THE PRICE PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF CLAIM. THE PERSON USING THE SOFTWARE BEARS ALL RISK AS THE THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

Some states do not allow the exclusion of the limit of liability for consequential or incidental damages, so the above limitation may not apply to you.

This agreement shall be governed by the laws of the State of Minnesota and shall inure to the benefit of David K. Rich, any successors, administrators, heirs and assigns. Any action or proceeding brought by either party against the other arising out of or related to this agreement shall be brought only in a STATE or FEDERAL COURT of competent jurisdiction located in Ramsey County, Minnesota. The parties hereby consent to personam jurisdiction of said courts.