

psMenu 1.54

PsMenu makes it easy to create BSPlayer menus from initialization files. In a few steps you will be able to control a playlist or select your favourite language and subtitles for playback. Complete your menu with some looped sound or music and a still picture (you cannot insert animated images).

This version of psMenu can run after installation on the user's hard disk or directly off a CD-ROM. The script can be placed in the root directory or in a folder (folder name must be in DOS format), or if you are using a CD-ROM the script must be stored in the same folders and must have the same name on each support.

PsMenu can be used along with Autoplay so that the menu and the movie will automatically load.

A program to create the script to build a menu is not included, but you can easily start from the sample scripts included in the documentation which can be modified to your needs.

If you run "psMenu.exe" directly, you will be requested to provide a valid script. The examples supplied should give you a good start. Scripts have a ".psmu" extension and both the program and the scripts share the same icon. A quick way to see and modify file extensions is to open a folder, select "tools", and then select "folder options".

PsMenu includes built-in multilanguage support. Language files are to be placed in the subdirectory pslang.

If PsMenu is unable to find the language files the program will select the language according to the local language information provided by the operating system (Italian by default, or English if the OS language is different from Italian.).

If you want you can also record macros. Right-click with your mouse to reach the context menu and record up to 99 menu choices. You can later re-execute them as you like. When you record a macro the actions related to the reproduction of a movie are not executed but stored in memory.

From version 1.5 you can save macros to disk. If you use the "salva come preferita" (save as favorites) function when opening the same script you can load the relevant macro and start it by pressing "CTRL L". It is also possible to save the macro in a separate file and then load it manually as the program will automatically check the compatibility between the macro and the script. Every macro can only be loaded together with the script it has been created with.

It is also possible to create a script log so as to let you edit it the way you like. You can activate this function through the parameter "/debug" or from the context menu in Explorer (right click on a psMenu script file). A file with the same name as the script will be created in the folder. If it is not possible to use the folder where the script is, the current user temp folder will be used.

Scripts: the basics

There are three main parts in a script: a header, the options, and the pages. Each section follows the rules of a Windows .INI file, so that you cannot put spaces before and after the equal sign ("="). Another section can be optionally added as well to describe how the program will display the information stored in the memory cells (see below).

Header [SCRIPT FOR PSMENU]

The header at present has only one directive: "Ver". The value is the version number of the script file. We will use the current version number, 1.54.

[SCRIPT FOR PSMENU]

Options [MAIN]

Here you can specify the colors you want to use, the sound you want to play as background music and an optional image, as well as the title and other general features of the script.

Directive	Descrizione	Default
Title	Main title for the menu. This command will be ignored if a graphic menu is loaded.	<i>No title</i>
Directory	Folder where the image, the music and the .bsi files are placed. Use a relative path in relation to the current directory (where the script is stored).	<i>Directory path where the script is placed</i>
WallPaper	Background image, shown on every page of the menu. <i>Obviously, even this command will be ignored if you choose a graphic menu.</i>	<i>No image. The menu will show a black background.</i>
tile	With a value set to 1, if an image is too small to fill the entire area it will be repeated over and over to fill the area. With a value set to 0 the program will centre the image and resize it if necessary; If the value is set to 2 psmenu will stretch the image to fill the screen. <i>Ignored if a graphic menu is chosen.</i>	<i>Set to 1 (tiling on)</i>
Sound	Background music played while browsing the menu. The sound file must have a RIFF wav header (*).	<i>No music/sound</i>
FontLighColor	Specifies the color of the menu items when the mouse pointer moves over them (highlight color). Its value must be a valid color name, in English (see list).	<i>Red</i>
FontNormalColor	Specifies the "normal" color of the menu items. In English (see list).	<i>Yellow</i>
TitlePageColor	Sets the color of the page title (each page can have a different title. See below). Its value must be a valid color name, in English (see list).	<i>Blue</i>
ShowBar	A value of 1 will show the menu as a standard window with its title bar, otherwise the menu will fill the screen.	<i>Set to 1</i>
BackColor	It's the background color of the page. It is shown when the wallpaper is missing, or when it doesn't fill the screen completely (in English, see color list).	<i>black</i>

DefaultMemory	<p>The script can store up to 200 memory cells, numbered starting at 0 (see below).</p> <p>This directive specifies the value needed to initialize them. All values must be delimited by commas and cannot contain blank spaces or punctuation marks. An empty value can be set using two commas without any spaces. For example: DefaultMemory=,4,,7 stands for: Memory(0)="" Memory(1)="4" Memory(2)="" Memory(3)="7"</p> <p>The other values (4 to 199) are empty ("").</p> <p>Please note that the index starts at zero.</p>	<i>Empty Memory (all cells are empty)</i>
RunBsPlayFromCD	<p>Specifies whether the program has to look for bsPlayer on the CD-ROM or not.</p> <p>If set to 1 psMenu will look for "bsplay.exe" in the main folder of the CD-ROM, or in the subfolder "exec", or in the folder set in <i>Directory</i> directive. With a value of 0 the program will look for bsPlayer on the user's hard disk (getting the information needed through the ".bsi" extension)</p>	<i>Set to 0</i>
CD	<p>Specifies the number of the CD where the script is stored. If the movie is spanned on more than one CD each CD has to have a different number.</p>	<i>Set to 1</i>
VisMemory	<p>Displays the contents of the specified cells. All values must be delimited by commas. Visualization can be customized (for further information see [FORMAT] section).</p>	<i>No memory cell displayed</i>
TrackAudio	<p>Specifies the number of audio streams of the movie. It is a value required by OpenBsi.</p>	<i>Set to 2</i>
IgnoreMaxCharBsPlay	<p>If bsPlayer is on the CD and the filename is not in DOS format, an optional second part will not start automatically. If this directive is set to zero the program will warn you with a possible error, or it will ignore the wrong format.</p>	<i>Set to 0</i>

HotSpot	If set to 1, this command tells psMenu to use graphic menus (like DVDs') instead of classics text menus. In order to use graphic menus you need to create your images with <i>Image Editor</i> .	<i>Set to 0</i>
Warning	He with the equal value to 1 psMenu warns the presence of a version of obsolete bsplayer, you otherwise don't show any message. (Useful when the new version of bsplayer doesn't work well)	<i>Set to 1</i>

Valid color names are:

Black, Maroon, Green, Olive, Navy, Purple, Teal, Gray, Silver, Red, Lime, Yellow, Blue, Fuchsia, Aqua, White.

(*) To convert regular mp3 files to mp3 files with a RIFF wav header you have to use a suitable program, i.e. WaveMp3 (<http://videoripper.cjb.net/>)

[MAIN]

Title=Collection

Directory=.

...

Formatting [FORMAT]

This is an optional section, only needed if you want to modify the way the memory contents are displayed. In the first part of this section you can specify color, transparency and time of visualization. In the second part you can modify the way the memory contents are actually displayed as a customized message.

ColorMemory	Specifies the font color for the message displayed. Its value must be a valid color name, in English (see list). Default value is "lime".
Transparent	Specifies whether to show the black background or not. If set to 1, the message will be displayed together with a part of the background image set in section [MAIN] (directive <i>Wallpaper</i>). Default value is 0.
Delay	Sets the length of time for the message to be displayed. Time must be given in seconds, with a maximum delay of 60 seconds. For a permanent visualization of the memory contents, set the value to 0. Default value is 5 seconds.
CellContents=DisplayedMessage ...	You can specify a list of customized messages to match the actual contents of the memory cells, to improve readability and visual interest. Do not insert blank spaces before and after the equal sign ("="). By default there is no list.

[FORMAT]

Delay=10

Transparent=1

ColorMemory=red

_SENG=English Subtitles
_SITA=Italian Subtitles
_LENG=English Track
_LITA=Italian Track

Pages [SHEET X]

As the central part of the menu, this section defines the menu items and the events associated with the items. There must be at least one page (SHEET 1), and each page can have a maximum number of twenty items. The blank space between the *Sheet* directive and the page number is optional but recommended.

Each page can be basically divided into two parts: one part is explanatory, while the other specifies the events/actions to carry out.

Comments can be added to the script, preceded by a semicolon (“;”). The use of other characters for a comment could be successful but you they could cause errors (it depends on your operating system) you would not probably be able to connect immediately to the comment itself.

TitlePage	Title of the current page. It will be displayed just beneath the main title. The title can be stored in Memory. <i>The page title won't be shown in graphic menus.</i>	
WallPaper	Background image for the page/sheet. <i>Obviously the directive will be ignored when you use graphic menus.</i>	<i>If missing the value set in Main will be used</i>
tile	It arranges the image as stated.. See “Main” for details.	<i>If missing the value set in Main will be used</i>
FontLighColor	Color of the menu item when the pointer moves over it (in English, see color list).	<i>If missing the value set in Main will be used</i>
FontNormalColor	Normal color for the menu item (in English, see list).	<i>If missing the value set in Main will be used</i>
TitlePageColor	Color for the title of the page (in English, see list).	<i>If missing the value set in Main will be used</i>
BackColor	Background color of the page, shown when there is no wallpaper or when it does not fill the screen (in English, see list)	<i>If missing the value set in Main will be used</i>

OnSoundEndXX	<p>Azioni da svolgere al termine del suono. Il primo parametro è il nome del file sonoro da monitorare, con il secondo parametro si indicano le azioni da svolgere, queste devono essere separata dal punto e virgola (;). I due parametri sono separati dalla virgola. Le azioni non possono essere di salto o di apertura file. Il file sonoro appena terminato sarà riavviato, a meno che non sia diversamente indicato nella lista delle azioni. E' possibile monitorare fino a 20 (00..19) file contemporaneamente.</p> <p>Esempio: OnEndSound00=suono.wav,OpenSheet(2);stopsound</p>	<p><i>Se manca ripete l'esecuzione del file sonoro appena terminato.</i></p>
Cache=0 or 1	<p>If Cache is set to 1 (default value) the default action will be carried out only once, otherwise it will be executed each time the page is opened.</p>	
DefaultAction=X,Y	<p>Action X is carried out after Y seconds. If the default action is associated with a menu item the menu item is highlighted and the action is executed according to the <i>Cache</i> directive. Please note that you need to specify the action, not the menu item, so that it is possible to execute an “invisible” action after a certain time. The action will then be carried out every time you open a page, regardless of the value set with <i>cache</i>.</p> <p>Default action is executed only after all pending commands have been carried out (see <i>Action command</i>). If Y>0, when an item is selected and you move the mouse pointer, the automatic choice process doesn't take place; the “invisible” actions are cancelled when any menu item is selected. Default action is cancelled when you leave a page as well (according to the value of Y), or when through the pending commands it is present a jump. If the value of X or Y is set to -1, then the directive is ignored. This is the only directive in which you can change both parameters with <i>Memory(X)</i>. See below.</p> <p><i>In case of graphic menus, all actions “invisible” for any Y>0.</i></p>	

ItemXX="Vedi il film" Action=Y	<p>HotSpot=0 It comprises two parts. The first part specifies the item number and the item itself, while the second part specifies the number of the action to carry out (Action=y). The two parts must be divided by “ ”. Items can be numbered from 0 to 19. They don’t need to be ordered but you cannot use the same number for more than one item. They always have to have double figures, so that the first item will be Item00, the second will be Item01 and so on. Text must be enclosed between inverted commas and cannot exceed 90 characters, otherwise the item will not be displayed. Action=Y simply refers to the command list which is to be executed when the user select the respective item. Y can only be a number (no alphabetical characters) and must match the directive ActionXX.</p> <p>HotSpot=1 The same as <i>hotspot=0</i> except for the string between inverted commas which has to be the number corresponding to the picture you want to select from the graphic menu, represented by the image given in <i>HotSpotImg</i>. Please note that if the value specified in <i>HotSpotImg</i> is missing the menu will be loaded as if <i>HotSpot=0</i>; this allows to create mixed menus, both graphic and textual.</p>
ActionXX=Command List (See below)	Action number must always be written in double figures. Action 99 is a default action and causes an error, so do not use it in your scripts . Numbering can be different for each sheet and there is no limit as for the number of lines. Commands in the list must be separated by a semicolon (“;”). Commands and are executed in order, with only one exception (see below).
AlignItem=0 or 1 or 2	Sets the alignment of the item text in a textual page. If set to zero (default value) the text will be centred, if set to 1 it will be right aligned, if set to 2 the text will be left aligned.
HotSpotImg=image.jpg	This directive is valid only if <i>HotSpot</i> is set to 1. It represents the image containing all the possible choices given to the user. The picture has to be placed <i>ImgSpot</i> sub-directory.
HotSpotMask=MaskImage.jpg	The image that you specify in <i>HotSpotImage</i> can contain some selectable pictures. These pictures may change when the pointer moves over them (or on a single mouse click). <i>HotSpotMask</i> defines the way they change. See “Image Editor”. The image has to be located in the <i>ImgSpot</i> subdirectory.
HotSpotDef=file.hmt	.hmt file contains the coordinates of all the pictures that will be displayed inside the “Menu” image, and the number which identifies each one of them. This file will be automatically generated by Image Editor.

Command List

This is a list of commands which can be used in the current version of psMenu:

File management

Openfile
OpenFile.Memory
OpenBsi
OpenBsi.stop
OpenOgm
StartPrg

Sounds

PlaySound
StopSound
beep

Messages

Message
Items.Rename
Label

Strings and numbers

Add
Random
Invert

Jumps

Action
ifJump

Other commands

Memory
NumberCD
SwapDisk
Close

OpenSheet

Syntax: *OpenSheet(X,N)*

X is the number of the page [SHEET X] that will be opened. The opened page will be the current page. N is optional; if set to one, on opening the sheet the "DefaultAction" directive will be ignored. If the parameter is missing or set to zero, the sheet will be opened following the default procedure.

Memory

Syntax: *Memory("string",X)*

"String" is a series of characters that will be stored into cell X. Always include the string between inverted commas. You can use up to 100 cells, numbered from 0 to 99. The value will be used later when the command OpenFile.Memory will be called. The same value can be called through *Memory(X)*. *Memory(X)* can be used as a command parameter: the value of cell X is the parameter that is actually passed to the command. The value of the "Memory" parameter can be set as the value of another "Memory" parameter; we will then have a case of indirect memory addressing.

The contents of the cells which can be optionally specified in *VisMemory* ([MAIN] section) will be displayed according to the [FORMAT] section. If this section is missing default formatting will be used.

Example:

Action01=Memory("2",0);OpenSheet(Memory(0))
will open page/sheet 2, because 2 is the value stored into cell 0.

If we store in memory the following:

Memory[1]:=10
Memory[10]:=20

the command *Memory(Memory(1))* will give the value of 20 as a result (as if it were *Memory(10)*, whose value is set to 20).

OpenFile

Syntax: *Openfile("FileName.ext",Y)*

"FileName.ext" is the name of the file you want to open, including the optional path (without the device letter). The search is carried out according to the directory path specified in the [MAIN] section. Y is the number of the CD on which the file is stored. If the programme can't find the file, it asks the user to insert the right CD. **You must pay your maximum attention to numbering, because the file will only be searched on the specified CD.** When the file is (about to be) opened the optional music will stop and it will be resumed when the menu pops up.

With the exception of bsplayer that could reside on CdRom all the other necessary programs to the opening of the file must be installed on the computer guest.

Warning. If you open a bsplayer file when the player is on the CD, the filename must be in DOS format (eight characters for the name, three characters for the extension, no spaces), otherwise an optional second part will not start automatically.

OpenFile.Memory

Syntax: *OpenFile.Memory("FileName.ext",Y,X0,..X29)*

X0...X29 specify the cell where the value has been stored. You can use from a minimum of 1 to a maximum of 30 elements, and each element is added to the filename before the extension ".ext". For the rest the command is the same as for *OpenFile*.

Example:

[MAIN]

DefaultMemory=,

...

[SHEET X]

...

Action00=Memory("_LENG",3);Memory("_SITA",4)

Action01=Memory("_SENG,4)

Action02=OpenFileMemory("movie.bsi",1,3,4)

...

If you execute Action02 the file "movie.bsi" will be loaded from the first CD because cells number 3 and 4 are empty (this is specified in the [MAIN] section, in *DefaultMemory*).

If you execute Action00 and then Action02 you will open the file "movie_LENG_SITA.bsi" because you first use the element stored in 3 and then the one in 4.

If you first execute Action01 and then Action02 you will open the file "movie_SENG.bsi" because element 3 is empty.

OpenBsi

Syntax: *OpenBsi("FileName.bsi",Y,Subtitles,Audio,Chapter)*

As distinct from *OpenFile* this command loads .bsi files only and it only works with bsPlayer ver. 0.84.484 or higher. Unfortunately bsPlayer does not provide any information as to its version when executed, so psMenu cannot check if the version of bsPlayer installed on the user's hard disk is able to receive the commands. You are therefore recommended to use

psMenu with the bsPlayer which is on the CD, or you can point out to the user the possibility of malfunction or failure.

Y is the CD number where the file "FileName.bsi" is located.

Subtitles is the number for the subtitles which is specified in the .bsi file. If it is set to zero the default value is used (bsi's *defaultSub*).

Audio is the number of the audio stream. If it is set to zero the default value is used (bsi's *Audio=X*).

Chapter is the number of the chapter you want to play. If this value is higher than the real number of chapters the program's behaviour can be unpredictable (the present version of bsPlayer plays the last chapter). If the value equals zero the default chapter is shown (the present version of bsPlayer plays the first chapter and the following ones on the inserted CD).

As for the chapter selection, this command differs from bsPlayer's. It is therefore necessary for you to try the chapter selection from psMenu as well. If you select a chapter but another chapter is played, this implies that the times supplied by the .bsi file are not correct. As to the chapters list, please remember that each part of the movie has to coincide with the beginning of a chapter. So, the first part must start at 00:00:00, while all the remaining parts must start with a chapter that was set in *StartTime*.

The *TrackAudio* directive must also be present in the [MAIN] section for the *OpenBsi* command to work properly.

Warning. If the bsplayer is on the CD, the filename must be in DOS format (eight characters for the name, three characters for the extension, no spaces), otherwise an optional second part will not start automatically.

OpenOgm

Syntax: *OpenOgm*("FileName.ogm",Y,Subtitle,Audio,Chapter,NumberOfSubtitles)

The command carries out the same actions (and has the same limitations) as *OpenBsi*, except for the fact that it works on .avi or .ogm or .mkv files. At the end of the file the next part is therefore not executed automatically.

Subtitle is the number of the subtitle which has to be shown. For example, if you set it to 1 you want to show the first subtitle. If set to zero no subtitle will be shown.

Audio is the number of the audio stream. Both zero and one refer to the first stream.

Chapter is the number of the chapter. Both zero and one refer to the first chapter. If *chapter* is greater than the actual number of chapters program behaviour will not be predictable (it depends on the version of bsplayer).

NumberOfSubtitles is the total number of subtitles in the .ogm or .mkv file.

Startprg

Syntax: *Startprg*("FileName.exe",Disk,wait,"parameters")

The command runs the executable file "FileName.exe". The program has to be located on one of the disks where the menu is. Location is given through "Disk" and search path is set in the "Directory" directive. "Wait" states whether the command has to wait the end of the program or not. Please note that if the command is followed by *close* the parameter *wait* will be ignored.

The option *wait* is ignored also from any programs.

"Parameters" is optional, he denotes the parameters to go on to the program.

PlaySound

Syntax: *PlaySound*("FileName.wav")

The command will play *filename.wav*. If no parameter is given the sound file provided in the [MAIN] section will be played.

StopSound

Syntax: *StopSound*

The currently playing music/sound is stopped.

Action

Syntax: *Action(XX)*

For all practical purposes, it is a jump instruction. It tells the programme to execute the actions specified through “ActionXX=”. Any other further command is ignored.

Please note that *Action(XX)* relates to the page (SHEET) which is currently opened, not to that in which it is defined. If you execute the jump to a page with a default action the instructions specified in *DefaultAction* are executed *after* the instructions defined in the jump (pending command), with two exceptions.

The *DefaultAction* has not executed if between the pending command is present the opening of a page (*openSheet*).

For example:

[SHEET 2]

...

Action00=Memory(“_LITA”,1);Action(01)

Action01=OpenFile.Memory(“movie.bsi”,1,1);OpenSheet(1)

...

[SHEET 3]

...

Action00=Memory(“_LENG”,1);OpenSheet(2);Action(01)

Action01=Memory(“_Director’s Commentary”,1)

The action executed by “Action(01)” in [SHEET 3] will be the one which has been defined in [SHEET 2] because the *OpenSheet* command changes the current page. It is obvious that “Action(01)” in [SHEET 2] refers to the same page because nothing has changed the current page.

[SHEET 1]

...

Action01=OpenSheet(2);Action(03)

[SHEET 2]

DefaultAction=2,0

...

Action02=OpenSheet(5);Action(00)

Action03=Memory(“LENG”,3)

As we have seen, “Action01” in [SHEET 1] executes actions which are specified in “Action03” in [SHEET 2], and the default action is executed only after these commands have been executed. In the following example “Action02” in [SHEET 2] (Default Action) will never be executed because a command opens another page.

[SHEET 1]

...

Action01=OpenSheet(2);Action(03)

[SHEET 2]

DefaultAction=2,0

...

Action02=OpenSheet(5);Action(00)

Action03=Memory("LENG",3);OpenSheet(4)

IfJump

Syntax: *ifjump(Parameter1,Parameter2,"Condition",XX)*

where:

“Condition” can be one the following:

“<” stands for « *Parameter1 less than Parameter2* »

“>” stands for « *Parameter1 greater than Parameter2* »

“=” stands for « *Parameter1 equal to Parameter2* »

“<>” stands for « *Parameter1 unequal to Parameter2* »

“<=” stands for « *Parameter1 less or equal to Parameter2* »

“>=” stands for « *Parameter1 greater o equal to Parameter2* »

XX is the action number to execute when condition is true.

If condition is false then the following command is executed.

If both parameters are numbers (integers) there will be a comparison between numbers, otherwise two strings will be compared.

All parameters, “Condition” included, can be stored in cells.

In case of jump this command behaves similarly to the unconditioned jump command *Action*.

Add

Syntax: *add(CellNumber,lead,Operator1,Operator2,...,OperatorN)*

This instruction sums all the operators and stores the result in the cell stated by *CellNumber*.. Operators can be integers (which can also be recalled through Memory). If the parameter “lead” equals 1 and the result is less than 10 (as a positive number) then the result will have a zero before the number (01,02,etc).

Besides summing up integers, this function can also join strings. The kind of operation to carry out is stated by the first operand. Strings must be enclosed between inverted commas (“”). If you sum strings you must provide the “lead” parameter, even though it will be uninfluential. Example:

add(10,1,2,3)

will store 05 in cell number 10

add(10,1,"pippo",2)

will store “pippo2” in cell number 10

add(10,0,2,"pippo")

will generate an error during the execution because it tries to sum the number 2 and the string “pippo”

Compatibility with psMenu 1.3

If the script version is 1.3 and the first operator is less than 10 and has a preceding zero (for example: 01,02, etc.) then the result will have a preceding zero if less than 10. In this case you have not to state the “lead” parameter.

Invert

Syntax: *Invert(CellNumber,Operator1)*

This directive calculates the opposite for the value stated in *Operator1* and stores it in *CellNumber*. *Operatore1* can only be an integer.

Random

Syntax: *Random(CellNumber,MaxValue,lead)*

Random is useful when you need a random integer. It calculates a number between 1 and *MaxValue* (1 and *MaxValue* included) and stores it in *CellNumber*. If *lead* is set to 1 and the result is less than 10 (and a positive number) then a zero will be added to the result.

Openbsi.stop

Syntax: *OpenBsi.stop*("FileName.bsi",*Y*,*Subtitle*,*Audio*,*Chapter*,*EndChapter*)

This directive is similar to *OpenBsi*, but only chapters between *Chapter* and *EndChapter* (excluded) will be shown. The moment reproduction will stop is calculated automatically and will be based on the duration of the chapters. *PsMenu* cannot detect the eventual shift in the point of reproduction within the chapter and it will then wait for the previously calculated time.

The *EndChapter* parameter is optional, if it is missing the movie will stop at the following chapter.

Beep

Syntax: *Beep*(*Number*)

It plays one of Windows' default sounds.

Number can be set to one of the following:

0 --> "Default Windows sound"

1 --> "Confirm" sound

Message

Syntax 1: *Message*("string")

Syntax 2: *Message*("string",*CellNumber*)

You can use the symbol \$\$ in "string" whenever you want to begin a new paragraph.

The first syntax shows the message *string* and waits for *OK* to be pressed (typical Windows message).

The second syntax shows the *string* - which is typically a question - and waits for the answer.

The answer can be yes (1) or no (0) and it will be stored in *CellNumber*.

Items.Rename

Syntax: *Items.Rename*("string",*KeyNumber*)

This instruction renames the key provided by *KeyNumber* in the current page. *String* has the same restrictions stated in *Items* (*Sheet* section).

Label

Syntax: *Label*("string",*blink*)

The *string* will be shown in the upper right corner, under the Memory messages. If *blink* is set to 1 the message will blink. If you do not want the message to appear you can provide a blank string.

Label2

Syntax: *Label2*("string",*blink*,*Color*,*Transparency*)

The *string* will be centred in the upper part of the page, just under the title. If *blink* is set to one the message will blink. You can use the symbol \$\$ to begin a new paragraph, and only the first occurrence will be used.

The string will be colored as stated (in English, as in the color list) on a transparent background if the *Transparency* parameter is set to one, or on a black background if the parameter is set to zero. If you want to avoid the message you can write a blank string.

NumberCD

Syntax: *NumberCD*(*CellNumber*)

This directive stores the currently used cdrom number in *CellNumber*.

SwapDisk

Syntax: *SwapDisk(DiskNumber)*

This command asks the user to change cdrom. The new cdrom is stated through *DiskNumber*. The program waits for the change to be carried out.

Close

Syntax: *Close*

This is a simple command to close a script. Any other further command is ignored.

All commands are executed in order except for *close* if it immediately follows one of the “File Management” commands, in which case the .bsi file is executed and the script is closed *at the same time*.

If the *Close* command is missing and you are using one of the OpenFile commands, psMenu keeps running waiting for the movie to finish. Priority level will be set to idle while waiting, in order not to interfere with other running programs.

See the enclosed examples if you want to improve your knowledge on scripts.

Report bugs and suggest improvements to psmenu@tiscali.it (Italian or elementary English). Do not forget to indicate your Operating System and always attach the script file that caused the error.