

**# 75 External procedure documentation**

Written by Todd Carper

February 29, 1988

This document describes how to use the following External procedures located in the file February.EXT:

*ChangeItemIcon*  
*ChangeParam*  
*ChooseFile*  
*ClearScreen*  
*ColorScreen*  
*DefinePopupMenu*  
*DiskInfos*  
*DocumentInfos*  
*FillMap*  
*FontList*  
*FramedButton*  
*GetFontID*  
*GetFontName*  
*GetMap*  
*GetPicture*  
*GetPopupMenu*  
*GetSerialValue*  
*GetString*  
*ManageMap*  
*OvalButton*  
*PictureInfos*  
*PopupMenu*  
*Quit4D*  
*ReadButton*  
*SetFndrInfo*  
*SquareButton*  
*SystemInfos*  
*Text2Pict*  
*YesNo*

You may also find this information by selecting the procedure and choosing Get Info from within 4D External Mover.

<u>Variable name</u>	<u>Type</u>
<i>IntVar</i>	Integer
<i>StrVar</i>	String
<i>LIntVar</i>	Long Integer
<i>TextVar</i>	Text



*PictVar*                      Picture  
*DateVar*                      Date

***ChangeItemIcon(IntVar1; IntVar2; IntVar3)***



**ChangeItemIcon** changes the icon of an item in a menu. *IntVar1* is the menu ID, *IntVar2* is the line ID, *IntVar3* is the resource Icon ID.

If the item does not exist, **ChangeItemIcon** does not nothing.

The icon put in the item is one whose resource ID is equal to *IntVar3*+256.

To delete an icon from an item, pass 0 in the *IntVar3* paramater.

Important:

This procedure is local to the current menu bar, if you change the menu bar by calling **MENU BAR**, the next time you use the current menu bar, it will be displayed as it is defined in Design and NOT with the change you had implemented with **ChangeItemIcon**.

### **ChangeParam**(*IntVar1*;*IntVar2*)

**ChangeParam** is used to change values in the 'CUST' resource of 4th Dimension. *IntVar1* specifies the integer in the resource, *IntVar2* specifies the new value.

See Acius Technical note #16 to get information about the 'CUST' resource.

### **ChooseFile**(*StrVar1*,*StrVar2*)

**ChooseFile** displays on the screen the standard dialog to open a document. *StrVar1* returns the path of the selected file. Only files with the types specified by *StrVar2* are shown.

If the user selects and opens a file, its full pathname is returned in the *StrVar* parameter. If the user clicks on the cancel button, an empty string is returned in *StrVar*.

If *StrVar* is an empty string, all types of documents are shown.

You specify the type(s) to be shown with a sequence of four characters which stand for the type of the file. You can specify up to 4 different types.

Example:

You want the user to be able to choose a MacPaint or a picture file. To do this, you write :

```
`The MacPaint™ file type is PNTG
```

```
`The Picture file type is PICT
```

```
ChooseFile(FileName;"PNTGPICT")
```

```
If(FileName#"")
```

```
  `a file has been choosen
```

```
else
```

```
  `The user has clicked on cancel
```

```
End if
```

Important



a) You cannot specify more than 4 types. If you specify more than 4 parameters, the others are ignored.

b) **ChooseFile** works like **SET CHANNEL(10;"")** or **SET CHANNEL(13;"")** except it does not open the file. After calling **ChooseFile**, you can open the file by using **SET CHANNEL(10;StrVar)** where *StrVar* contains the complete path of the file.

**ClearScreen(IntVar)**



**ClearScreen** fills the current window with the pattern specified by the numeric value of *IntVar*. *IntVar* may be in the range of 1 to 38 and specifies one of the 38 system patterns. (See Inside Mac for further info)

### **ColorScreen**(*IntVar*)

After calling **ColorScreen**, *IntVar* is equal to 1 if the system is running with Color QuickDraw, else it is equal to 0.

### **DefinePopupMenu**(*LIntVar*; *StrVar1*; *StrVar2*; *IntVar*)

Using **DefinePopupMenu**, you can define a pop up menu installed in a layout.

*LIntVar* is the external area variable.

You give the title of the menu in *StrVar1*. You specify the items in *StrVar2*. You can specify two kinds of pop up menus. If *IntVar* is equal to 0, the menu title is displayed with a frame. If *IntVar* is equal to 1, the title is displayed with a shadow frame.

To specify the items of the menu, separate each with a semicolon, for example: "Line1;Line2;Line3;;Quit". An empty string between 2 semicolons means an inactive separation line. You can use all metacharacters of the Menu manager in the string to specify marks, styles variations, icons.

Note:

ShortCuts are not supported in a popup menu.

### **DiskInfos**(*StrVar*; *DateVar1*; *DateVar2*; *DateVar3*; *LIntVar1*; *LIntVar2*; *LIntVar3*; *LIntVar4*; *LIntVar5*; *LIntVar6*)

*Var1*... is an alphanumeric variable

*Var2*... *Var4* are date variables

*Var5*... *Var10* are numeric variables

DiskInfos returns information about the disk whose name is specified by *StrVar*. To get information about the disk of the database, pass an empty string in *StrVar*. To specify another disk, pass its name followed by a semicolon in *StrVar*.

*StrVar*, contains the name of the disk

*DateVar1*, contains the date of the initialization.

*DateVar2*, contains the date of the last modification.

*DateVar3*, contains the date of the last backup.

*LIntVar1*, contains the time of the initialization expressed in seconds since midnight.

*LIntVar2*, contains the time of the last modification expressed in seconds since midnight.

*LIntVar3*, contains the time of the last backup expressed in seconds since midnight.

*LIntVar4*, contains 1 if the disk is an HFS disk, 0 if the disk is a MFS disk.

*LIntVar5*, contains the number of files found on the disk.

*LIntVar6*, contains the free space in bytes on the disk.

If the disk does not exist, *LIntVar4* is equal to -1.



**DocInfos(StrVar)**

**DocInfos** returns information about the document whose name is specified by *StrVar*.

*StrVar* contains the full pathname of the document

The following 4th Dimension variables are created:

F.Type contains the File Type.

F.Creator contains the File Creator.

F.DaLogLen contains the logical size of the data fork which means the number of written bytes in the file.

F.DaPhyLen contains the physical size of the data fork which means the allocated space on the disk for the file.

F.ResLogLen, like F.DaLogLen but for the resource fork.

F.ResPhyLen, like F.DaPhyLen but for the resource fork

Note:

The logical size given by the Finder is equal to F.DatogLen+F.ResLogLen. The physical size given by the Finder is equal to F.DaPhyLen+F.ResPhyLen.

F.CreDate contains the date of the creation of the document.

F.ModDate contains the date of the last modification.

F.CreTime contains the time of the creation expressed in seconds since midnight.

F.ModTime contains the time of the last modification expressed in seconds since midnight.

Note:

If the document does not exist, F.DaLogLen is equal to -1.

**FillMap(LIntVar;PictVar)**

See **ManageMap**

**FontList**

This procedure does not require parameters.

After calling **FontList**, you get the list of all available fonts of the current system file in the array of 4th Dimension variables **Fonts1**, **Fonts2**, ...

**Fonts0** contains the number of fonts.

**FramedButton**

**FramedButton** works like **SquareButton** except that the button is highlighted by only a frame.

**GetFontID(StrVar;IntVar)**

After calling **GetFontID**, *IntVar* contains the Font ID of the font whose name is specified by *StrVar*.



**GetFontName**(*IntVar*; *StrVar*)

After calling **GetFontName**, you get in *StrVar* the name of the font whose ID is specified by *IntVar*.

**GetMap**(*LIntVar*; *IntVar*)

See *ManageMap*

**GetPicture**(*IntVar*; *PictVar*)

After a call to **GetPicture**, you get in *PictVar* the picture stored as a resource whose resource ID is specified by *IntVar*. If the picture is correctly read, *IntVar* stays unchanged, if the picture does not exist or cannot be read, *IntVar* is equal to 0.

**GetPopMenu**(*LIntVar*; *IntVar*; *StrVar*)

You call **GetPopMenu** in each **During** phase of a layout procedure of a layout which contains a popup menu.

If the **During** call is due to an action of the user in the menu. *IntVar* contains the ID of the selected item and *StrVar* contains the text of the item.

If the user has not selected an item or if the **During** call is due to another event, *IntVar* is equal to 0 and *StrVar* is an empty string.

**GetSerialValue**(*IntVar1*; *IntVar2*; *IntVar3*)

**GetSerialValue** displays on the screen a dialog which allows you to choose the configuration of a serial port including modem or printer port, speed, parity, protocol and so on...

If the user validates the dialog, *IntVar3* contains 1, if not *IntVar3* contains 0.

If the user validates the dialog you get, respectively in *IntVar1* and *IntVar2*, the values that you have to pass to **SET CHANNEL** to process the selection of the user. If the user does not validate the dialog, *IntVar1* and *IntVar2* are not modified.

If, when you call **GetSerialValue**, *IntVar1* and *IntVar2* contain significant values, the procedure checks corresponding boxes and buttons to reflect the values.

Note:

If you pass the null value in *IntVar2*, **GetSerialValue** displays these default values : 9600 baud, No parity, 8 data bits, 1 stop bit.

**GetString**(*StrVar*; *IntVar1*; *IntVar2*)

After a call to **GetString**, *StrVar* contains the string stored in the 'STR#' resource whose resource ID is specified by *IntVar1* and whose location in the list is specified by *IntVar2*. If the string is not found an empty string is returned.



### **ManageMap**

The routines **ManageMap**, **GetMap**, **FillMap** allow you to manage a multi-valued picture area. The user clicks on one picture to select it.

#### **FillMap**(*LIntVar*; *PictVar*)

*LIntVar* contains the handle to the external area.

*PictVar* is a picture expression

**FillMap** adds in the area specified by *LIntVar*, the picture specified by *PictVar*.

#### **GetMap**(*LIntVar*; *IntVar*)

*LIntVar* contains the handle to the external area.

Call **GetMap** in the **During** phase of the layout procedure to check if the user has clicked on a picture. If so, you get the ID of the picture in *IntVar*.

### **ManageMap**

This routine does not require parameters. **ManageMap** is the routine associated with the external area.

### **OvalButton**

**OvalButton** works like **SquareButton** except that the button is highlighted by an oval.

#### **PictureInfos**(*PictVar*; *IntVar1*; *IntVar2*; *IntVar3*; *IntVar4*)

**PictureInfos** returns in variables the size and the location of the picture specified by *PictVar*.

*IntVar1* is equal to the horizontal coordinate of the top left corner of the picture.

*IntVar2* is equal to the vertical coordinate of the top left corner of the picture.

*IntVar3* is equal to the width of the picture.

*IntVar4* is equal to the height of the picture.

All values are expressed in pixels.

### **PopMenu**

This procedure does not require parameters

**PopMenu** allows you to use Popup menus in 4th Dimension layouts. To do so, you have to create an external area in a layout and give **PopMenu** as the associated routine. To define the content of the Popup menu you use the **DefinePopMenu** routine. To get the ID and text of the item selected by the user, you use **GetPopMenu**.





**Quit4D(IntVar)**

**Quit4D** allows you to automatically quit a 4th Dimension database. The *IntVar* is used to specify what you want to do.

*IntVar*=1, Quit displays the "Are you sure?" dialog, then, if user clicks the OK button, flushes and ejects all disks and shuts down the Macintosh.

*IntVar*=2, Quit displays the standard dialog to open another Macintosh application.

*IntVar*=3, Quit displays the "Are you sure?" dialog, then, if user clicks the OK button, goes back to the Finder.

*IntVar*=11, like *IntVar*=1 but without displaying the "Are you sure?" dialog

*IntVar*=12, like *IntVar*=2 but when the user quits the launched application he will go back to 4th Dimension and not to the Finder.

*IntVar*=13, like *IntVar*=3 but without displaying the "Are you sure?" dialog.

If your database is running under multi-user mode 2 and 12 are not supported. 1 and 11 work respectively like 3 and 13. **Quit4D** works correctly under multi-user mode only if the database has been opened with a password with an associated startup procedure.

**ReadButton(LIntVar;IntVar)**

*LIntVar* is the handle to an external area

You must call **ReadButton** in the **During** phase of a layout procedure for each picture button you have installed in a layout.

If the call to the **During** phase is due to a click in a button, *IntVar* is equal to 1 else it is equal to 0.

**SetFndrInfo(StrVar1;StrVar2;StrVar3)**

**SetFndrInfo** sets the document whose name is equal to *StrVar1* to the type and the creator specified by *StrVar2* and *StrVar3*. If the document does not exist, **SetFndrInfo** does nothing.

**SquareButton**

This procedure does not require parameters.

This procedure manages an external area in an 4th Dimension layout and simulates an invisible button which is highlighted by an invert. You can get the value of the button by calling **ReadButton**. The button is 1 if it has been clicked, 0 if not.

**SystemInfos(StrVar1;StrVar2;StrVar3;StrVar4;StrVar5;StrVar6;LIntVar1;LIntVar2;LIntVar3)**

By calling **SystemInfos**, you get useful information about the system environment of your database. See Inside Mac...for more info.

*StrVar1*, contains a string equal to the system version.

*StrVar2*, contains a string equal to the Finder version.

*StrVar3*, contains a string equal to the version of the current selected printer.



*StrVar4*, contains a string equal to the version of the Macintosh.

*StrVar5*, contains a string equal to the current file system used ( MFS or HFS)

*StrVar6*, contains a string equal to the version of CPU (68000 or 68020)

*LIntVar1*, contains the memory size in Kbytes

*LIntVar2*, contains the free memory in bytes.

*LIntVar3*, contains the size in bytes of the cache memory.



**Text2Pict**(TextVar;PictVar;IntVar1;IntVar2;IntVar3;IntVar4;Var2)

**Text2Pic** returns a picture in *PictVar* which is the graphical representation of the text specified by *TextVar*.

*TextVar* can have up to 32000 characters. The others parameters are used as follows :

*IntVar1* specifies the justification: 0=Align Left, 1=Centered,-1=Align right.

*IntVar2* specifies the font.

*IntVar3* specifies the size of characters ( from 1 to 128)

*IntVar4* specifies the style variations: 0=Plain, 1=Bold, 2=Italic, 4=Underline, 8=Outline, 16=Shadow, 32=Condensed, 64=Extended. You can add values to specify several variations.

*IntVar5*:

Before the call: transfer mode of the picture

After the call : 0 if the picture has been created, -1 if the picture has not been created.

**YesNo**(StrVar;IntVar)

**YesNo** displays a dialog with the message specified by *StrVar*. There are 3 buttons in the dialog : Yes, No and Cancel. If the user clicks in Yes, *IntVar* is equal to 1, if the user clicks in No, it is equal to 2, if the user clicks in Cancel it is equal to 3.

